FLUENT 全攻略

流體中文網傾情奉獻

雷鋒精神永放光芒！

2005 年 3 月 5 日

VISIT…

LANZAROTE Caliente.com

版權聲明

本書乃周華站長、孫爲民、徐麗、宋劍的個人工作成果，僅供流體中文網網友下載交流之用，請下載后 24 小時內刪除。本網對書中內容不承擔任何法律責任，請謹慎使用！

祝大家身體健康，萬事如意！

2005 年 3 月 5 日星期六

紀年學習雷鋒四十二周年

第一篇 FLUENT 基础知识

第一章 FLUENT 软件介绍

FLUENT 软件是目前市场上最流行的 CFD 软件，它在美国的市场占有率达到 60%。在我们进行的网上调查中发现，FLUENT 在中国也是得到最广泛使用的 CFD 软件。因此，我们将在这本书中为大家全面介绍 FLUENT 的相关知识，希望能让您的 CFD 分析工作变得轻松起来。

用数值方法模拟一个流场包括网格划分、选择计算方法、选择物理模型、设定边界条件、设定材料属性和对计算结果进行后处理几大部分。本章将概要地介绍 FLUENT 软件的以下几个方面：

（1）FLUENT 软件的基本特点。
（2）FLUENT、GAMBIT、TECPLOT 和 EXCEED 的安装和运行。
（3）FLUENT 的用户界面。
（4）FLUENT 如何读入和输出文件。
（5）FLUENT 中使用的单位制。
（6）如何规划计算过程。
（5）FLUENT 的基本算法。

1.1 FLUENT 软件概述

1.1.1 网格划分技术

在使用商用 CFD 软件的工作中，大约有 80%的时间是花费在网格划分上的，可以说网格划分能力的高低是决定工作效率的主要因素之一。FLUENT 软件采用非结构网格与适应性网格相结合的方式进行网格划分。与结构化网格和分块结构网格相比，非结构网格划分便于处理复杂外形的网格划分，而适应性网格则便于计算流场参数变化剧烈、梯度很大的流动，同时这种划分方式也便于网格的细化或粗化，使得网格划分更加灵活、简便。

FLUENT 划分网格的途径有两种：一种是用 FLUENT 提供的专用网格软件 GAMBIT 进行网格划分，另一种则是由其他的 CAD 软件完成造型工作，再导入 GAMBIT 中生成网

格。还可以用其他网格生成软件生成与 FLUENT 兼容的网格用于 FLUENT 计算。可以用于造型工作的 CAD 软件包括 I-DEAS、Pro/E、SolidWorks、Solidedge 等。除了 GAMBIT 外，可以生成 FLUENT 网格的网格软件还有 ICEMCFD、GridGen 等等。

FLUENT 可以划分二维的三角形和四边形网格，三维的四面体网格、六面体网格、金字塔型网格、楔型网格，以及由上述网格类型构成的混合型网格。

1.1.2 软件的灵活设置

FLUENT 的内核部分是用 C 语言写成的，软件界面则是用 LISP 语言的一个分支 Scheme 语言写成的。因为 C 语言在计算机资源的分配使用上非常灵活，所以 FLUENT 也在这方面拥有很大的灵活性，并可以在“客户/服务器”模式下进行网络计算。而 LISP 类型的语言允许高级用户通过编制宏和自定义函数改变软件的外观，使用户在使用中可以根据自己的喜好定制界面，这点是 FLUENT 软件的一个显著特色。

1.1.3 软件的基本构成

FLUENT 软件包中包括以下几个软件：

（1）FLUENT 求解器——FLUENT 软件的核心，所有计算在此完成。

（2）prePDF——FLUENT 用 PDF 模型计算燃烧过程的预处理软件。

（3）GAMBIT——FLUENT 提供的网格生成软件。

（4）TGRID——FLUENT 用于从表面网格生成空间网格的软件。

（5）过滤器——或者叫翻译器，可以将其他 CAD/CAE 软件生成的网格文件变成能被 FLUENT 识别的网格文件。上述几种软件之间的关系如图 1-1 所示。

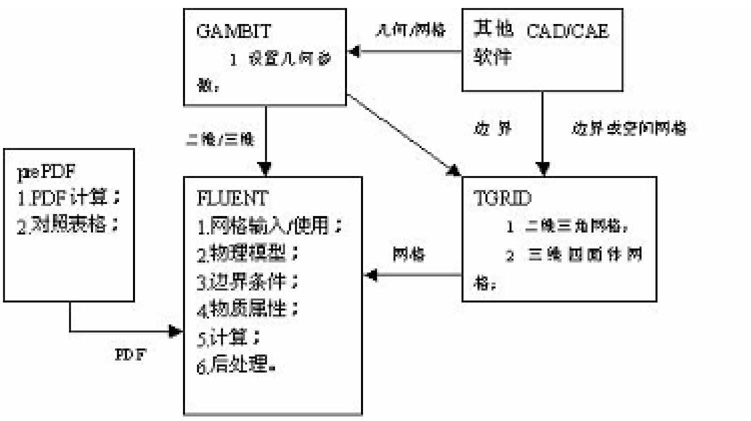


图 1-1 FLUENT 软件各组件之间的关系

GAMBIT 即可以生成用于 FLUENT 计算的网格，也可以生成其他一些 CAE 软件的网

格。一旦网格文件被输入到 FLUENT 中，剩下的工作就由 FLUENT 完成。这些工作包括设定边界条件和物质属性、调整网格、进行计算、对结果进行后处理并显示结果等等。

1.1.4 计算类型及应用领域

FLUENT 可以计算的流动类型包括：

（1）任意复杂外形的二维/三维流动。

（2）可压、不可压流。

（3）定常、非定常流。

（4）无粘流、层流和湍流。

（5）牛顿、非牛顿流体流动。

（6）对流传热，包括自然对流和强迫对流。

（7）热传导和对流传热相耦合的传热计算。

（8）辐射传热计算。

（9）惯性（静止）坐标、非惯性（旋转）坐标下中的流场计算。

（10）多层次移动参考系问题，包括动网格界面和计算动子/静子相互干扰问题的混合面等问题。

（11）化学组元混合与反应计算，包括燃烧模型和表面凝结反应模型。

（12）源项体积任意变化的计算，源项类型包括热源、质量源、动量源、湍流源和化学组分源项等形式。

（13）颗粒、水滴和气泡等弥散相的轨迹计算，包括弥散相与连续项相耦合的计算。

（14）多孔介质流动计算。

（15）用一维模型计算风扇和换热器的性能。

（16）两相流，包括带空穴流动计算。

（17）复杂表面问题中带自由面流动的计算。

简而言之，FLUENT 适用于各种复杂外形的可压和不可压流动计算。

1.1.5 系统要求

硬件要求:

（1）CPU：采用 Intel 的奔腾系列或 AMD 的 Athlon 系列 CPU。

（2）显示卡：分辨率为 1024x768（或更高），颜色最低要求 256 色，推荐使用 16 位（或 24 位）真彩色。

（3）内存：最低 128 兆内存，推荐使用 512 兆以上的内存。

（4）鼠标：推荐使用三键鼠标，FLUENT 中有些操作需要鼠标中键。

（5）光驱：最低需要 CD-ROM，FLUENT 需要光盘安装。

（6）网卡：需要安装以太网卡，FLUENT 需要网络验证。

（7)硬盘：FLUENT 需要 400 兆，GAMBIT 需要 55 兆，EXCEED 需要 105 兆，TECPLOT 需要 30 兆，一共需要 590 兆硬盘空间。如果需要使用用户定义函数（UDF）编译功能，还需要安装微软的 Visual C++，至少需要 200 兆硬盘空间，再加上为计算项目预留空间，所以推荐在硬盘上预留 5G 以上的硬盘空间用于 FLUENT 计算。

软件要求:

（1）操作系统：UNIX、LINUX、WINDOWS 2000/XP 等等。

（2）网络协议：安装 TCP/IP 协议。

（3）编译环境：如果使用 UDF 编译功能，需要安装 Visual C++标准版，最低要求是采用“最小安装”设置。

（4）EXCEED X server：GAMBIT 运行需要此环境，如果在 GAMBIT 中使用 OpenGL 图形模式，则需要安装 EXCEED 3D。EXCEED 3D 需要单独从 EXCEED 开发商 Hummingbird 软件公司购买。

1.2 软件安装与启动

与 FLUENT 配套使用的软件包括前处理软件 GAMBIT、后处理软件 TECPLOT 和模拟 UNIX 环境的 EXCEED。下面分别介绍这几种软件的安装和启动过程。

1.2.1 FLUENT 的安装与启动

FLUENT 的安装

（1）首先在 C 盘上创建目录 C:\flexlm，然后将软件公司提供的 license.dat 文件拷贝到这个目录下。

（2）将 FLUENT 的安装光盘放入光盘驱动器，然后双击 setup.exe 文件启动安装程序，设置好安装目录、工作组名称等选项后，安装程序开始向硬盘拷贝文件。

（3）文件拷贝结束后，在弹出的工作组窗口中双击运行 set environment，设置好环境变量，再将 license（软件使用许可证）服务器的地址设置完毕，重新启动计算机，就可以启动 FLUENT 了。

这里需要说明的是：FLUENT 与我们以往使用的软件不同之处是其安装过程中不需要使用安装密码，而是在每次启动软件之前，通过互联网在 license 服务器上进行软件合法性验证，只有通过验证才能使用。

单精度和双精度求解器

在所有的操作系统上都可以进行单精度和双精度计算。对于大多数情况来说，单精度计算已经足够，但在下面这些情况下需要使用双精度计算：

（1）计算域非常狭长（比如细长的管道），用单精度表示节点坐标可能不够精确，这时需要采用双精度求解器。

（2）如果计算域是许多由细长管道连接起来的容器，各个容器内的压强各不相同。如果某个容器的压强特别高的话，那么在采用同一个参考压强时，用单精度表示其他容器内压强可能产生较大的误差，这时可以考虑使用双精度求解器。

（3）在涉及到两个区域之间存在很大的热交换，或者网格的长细比很大时，用单精度可能无法正确传递边界信息，并导致计算无法收敛，或精度达不到要求，这时也可以考虑采用双精度求解器。

启动 FLUENT

以 WINDOWS NT 为内核的操作系统包括 WINDOWS 2000 和 WINDOWS XP，其启动方式有两种：

（1）从 WINDOWS 的开始菜单中进行启动，即顺序点击：

开始 -> 程序 -> Fluent Inc. -> FLUENT 6.1

就可以启动 FLUENT。

（2）从 DOS 终端窗口启动，即在命令行中：

1）键入“fluent 2d”，启动二维单精度计算。

2）键入“fluent 3d”，启动三维单精度计算。

3）键入“fluent 2ddp”，启动二维双精度计算。

4）键入“fluent 3ddp”，启动三维双精度计算。

如果想启动并行计算模式，可以在上述 4 个命令后面加上-tx 参数，其中 x 是并行计算的 CPU 数量，例如键入“fluent 3d –t3”意思是在三个处理器上运行三维计算。

1.2.2 GAMBIT 的安装与启动

GAMBIT 的安装

GAMBIT 的安装与 FLUENT 类似。在安装完 FLUENT 后，点击进入安装盘上的 GAMBIT 目录，找到 setup.exe 文件，双击启动这个安装文件，即开始安装 GAMBIT。经过与 FLUENT 安装类似的工作目录选择、工作组名称选择等设置后，安装程序开始向硬盘拷贝程序，拷贝结束后，安装过程就结束了。

GAMBIT 的启动

GAMBIT 需要在 UNIX 系统下运行，因此为了在 WINDOWS 下运行还需要安装 EXCEED 为 GAMBIT 提供一个模拟的 UNIX 环境。

GAMBIT 的启动方式有两种：一种是在 WINDOWS 系统的“开始”菜单中的“运行”栏里面直接键入命令 gambit –id project（project 是可以任意给定的项目名称）。另一种是在 DOS 环境中，在命令行中键入上述命令启动 GAMBIT。

1.2.3 TECPLOT 的安装与启动

TECPLOT 的安装与 WINDOWS 下的常见软件安装一样：

（1）在安装文件所在目录里，双击 setup.exe 文件开始安装。

（2）按照提示点击 NEXT（下一步）按钮，在用户许可协议页面中选择 I agree（我同意）按钮，表示同意上述协议。

（3）在许可代码栏中填入代码。

（4）选择安装目录。

（5）开始拷贝文件。

（6）文件拷贝结束后，重新启动计算机。

（7）双击桌面上的图标，或者在“开始”菜单中找到 TECPLOT，并点击启动软件。

TECPLOT 需要大约 30 兆的硬盘空间，可以在 WINDOWS 系列操作系统下正常运行。

1.2.4 EXCEED 的安装与启动

EXCEED 的安装

EXCEED 的安装也与常见软件类似：

（1）假设 F 盘是光驱，进入 F:\EXCEED\XSERVER 目录，双击启动 setup.exe，软件开始对安装环境进行初始化。

（2）软件提示是否同意用户协议，选择 I agree（我同意）进入下一步。

（3）指定安装主目录，比如 D:\Program Files\Exceed.nt。

（4）指定用户目录，比如 D:\Program Files\Exceed.nt\user。

（5）安装程序开始向硬盘拷贝文件，结束时会提示重新启动计算机。

（6）重新启动计算机，就可以使用 EXCEED 了。

因为 EXCEED 仅用于为 GAMBIT 提供一个模拟的 UNIX 环境，所以这里不对 EXCEED 做过多的介绍，详细介绍可以参考 EXCEED 的用户手册。

EXCEED 的启动

在运行 GAMBIT 前，需要先启动 EXCEED，启动方法是在 WINDOWS 的开始菜单中依次选择：

开始 -> 程序 -> EXCEED -> EXCEED

在 GAMBIT 的最新版本中，可以用批命令的方式同时启动 EXCEED 和 GAMBIT。

1.3 FLUENT 用户界面

FLUENT 同时采用了图形用户界面和文字用户界面二种界面形式进行操作，下面分别进行介绍。

1.3.1 图形用户界面

FLUENT 采用 WINDOWS 风格的图形用户界面，其界面类型主要分为下列四种：

（1）主控窗口。

（2）控制参数面板。

（3）消息窗口。

（4）图形显示窗口。

下面分别进行介绍。

主控窗口

启动 FLUENT 后首先进入的窗口就是主控窗口，见图 1-2。主控窗口分为菜单栏和文字信息窗口两个部分。用户与 FLUENT 软件之间的互动可以用两种方式进行，一种是在下拉菜单中直接选择相应的菜单命令进行操作，或者在弹出的参数控制面板上进行参数选择操作，同时系统通过消息窗口将系统信息直接反映给用户；另一种是在文字信息窗口中键入命令进行操作，系统信息也以文字信息的方式反映在文字信息窗口中，就象在 DOS 环境中一样。与微软公司的 WORD 等软件一样，当窗口中信息量超过一个页面的容量时，页面内容将自动上移，同时窗口右端的滚动条被激活，用户可以通过移动滚动条上的滑块查看滚动出页面的信息。

在命令执行过程中，可以按组合键 Ctrl+C 终止命令的执行。

菜单命令的执行除了可以用鼠标左键点选执行的方式外，还可以通过热键方式进行。比如点击打开 File 菜单的动作，既可以通过将光标移动到 File 上，再按左键点击打开的方式完成，也可以通过热键方式完成。所谓热键方式就是用键盘操作代替鼠标操作的方式，比如上述打开 File 菜单的动作可以用组合键 Alt+F 来完成，而不必移动鼠标。请注意在可以使用热键打开的菜单项中带下划线的字母就是打开这个菜单的“热键”，比如 File 菜单中

字母 F 下面带有下划线，即表示可以通过按组合键 Alt+F 的方式打开。同样，在 Grid 菜单项中，字母 G 下面带有下划线，表示可以通过按 Alt+F 组合键的方式打开，其他菜单与此相同。

在文字信息窗口中可以对显示的信息、命令进行剪切、拷贝、粘贴、删除等编辑操作，方法与微软的 OFFICE 系列软件中一样，即：

（1）首先将光标移动到要进行编辑区域的一角。

（2）按下鼠标左键并拖动到编辑区域的另一角，此时编辑区域被反白显示。

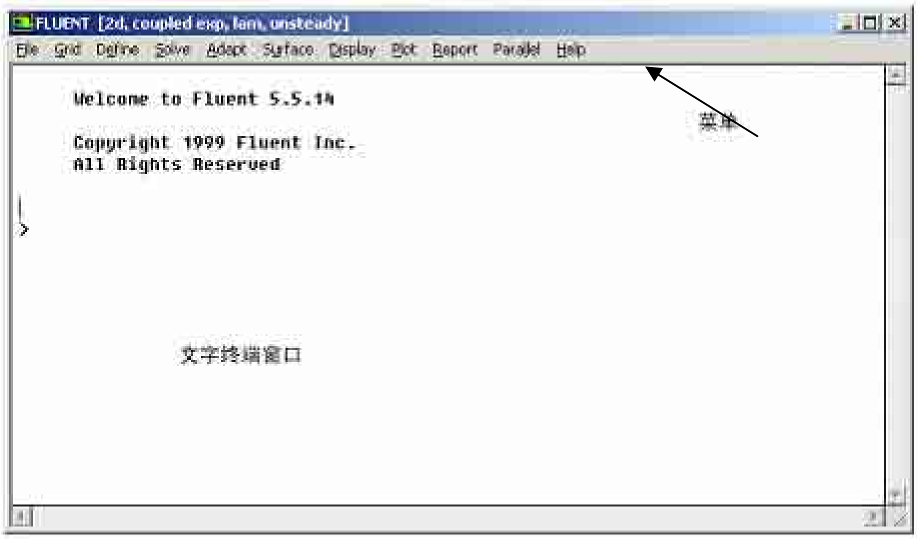


图 1-2 FLUENT 主控窗口

（3）将光标移动到反白显示区域，再按右键弹出编辑操作面板。

（4）在面板上选择剪切、拷贝或删除[1]。

（5）将光标移动到剪切或拷贝的目标位置。

（6）按右键弹出编辑操作面板。

（7）选择粘贴，则完成剪切或拷贝操作。

控制参数面板

控制参数面板类似于一个表格，是 FLUENT 软件中用于输入控制参数的主要形式之一，图 1-3 是用于网格显示操作的一个控制参数面板。

这个面板中包含了一些常见的可视化操作组件，如选择按钮、单选按钮、多选单和文本输入框，以及可以打开其他面板的按钮，如图中的 Display（显示）按钮和 Colors…（颜色）按钮，以及全选按钮和不选按钮。选择按钮和单选按钮被点击选中后，表示在计算中将采用相应的参数设置，再次点击则可以取消该项设置。多选单用于可以进行多重选择的

[1] 若进行删除操作，则到这一步编辑即告结束

场合。具体参数的设置需要人工决定时，则可以使用文本输入框输入相应的参数。点击全选按钮可以将多选框中所有选项选中，点击不选按钮则可以取消所有选择。

除了图 1-3 中所示的组件外，FLUENT 中还使用下拉菜单（图 1-4a）和单选菜单（图 1-4b）进行单一选择操作。FLUENT 除了使用直接的文字输入进行参数设置，还使用整数增减框（图 1-5a）和滑尺（图 1-5b）进行数字的输入操作。如同其他 WINDOWS 风格的软件一样，FLUENT 中也使用“标签”组件，以便在同一个面板中选择不同种类或群组的输入操作（图 1-6）。

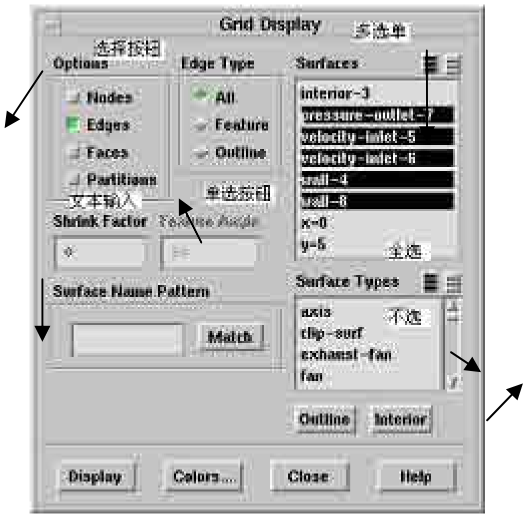


图 1-3 控制参数面板



图 1-4a 下拉菜单

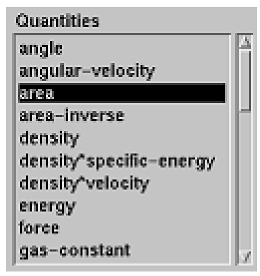


图 1-4b 单选菜单

整数增减框通过其右端的增减按钮来控制数值变化，数值显示在方框中。滑尺则通过拖动滑块完成数值输入工作，其数值显示在滑块上方。



图 1-5a 整数增减框



图 1-5b 滑尺

用鼠标点击不同的标签，则相应的选项就显示在标签下方，输入操作就可以开始进行。

在一项输入操作完成后，可以点击其他标签进行其他输入操作。比如在图 1-6 中，我们可以首先选择 Thermal（热力学）标签进行与热力学参数设置相关的操作，然后可以点击 Momentum（动量）标签进行与动量参数相关的的操作，等等。



图 1-6 标签

FLUENT 中的 OK、APPLY（应用）等按钮表示在计算中采纳面板中的输入值。CLOSE（关闭）按钮表示关闭面板，CANCEL（取消）按钮表示取消所有设置并关闭面板，HELP（帮助）按钮则进入在线帮助。这些常规组件的使用方法与其他 WINDOWS 风格软件一样，关键是要在理解问题的基础上进行选择，其操作则为大家所熟知，故不在此赘述。

消息窗口

在 FLUENT 的运行过程中，系统会根据需要弹出一些消息窗口，向用户提供系统信息，或提示用户进行一些简单的操作。

图 1-7a 是一个出错信息窗口，用户阅读过信息后可以点击 OK 键关闭窗口。



图 1-7a 系统消息



图 1-7b 工作信息

图 1-7b 是一个工作信息窗口，这个窗口告诉用户系统正在进行的工作信息。

图 1-9 是文件选择窗口，系统弹出这个窗口让用户选择要进行读写操作的文件。文件选择窗口的外观在大多数情况下与图 1-9 中的窗口类似，只是在选择绘制 XY 曲线文件时有所不同。

文件选择窗口的使用方法如下：

（1）进入工作目录。

在 Filter（过滤器）一栏中将完整的工作目录输入进去，然后点击 Filter 键，或在键盘上按回车键。也可以直接在下面的 Directories（文件夹）栏中双击相应的目录都可以进入相应的工作目录。注意在 Filter 栏中可以指定需要显示的文件类型，比如用星号“*”显示所有文件，用“*.dat*”显示扩展名为.dat 的文件等等。

与 DOS 系统相类似，Directories 中的单点“.”代表当前目录，双点“..”代表上级子目录。

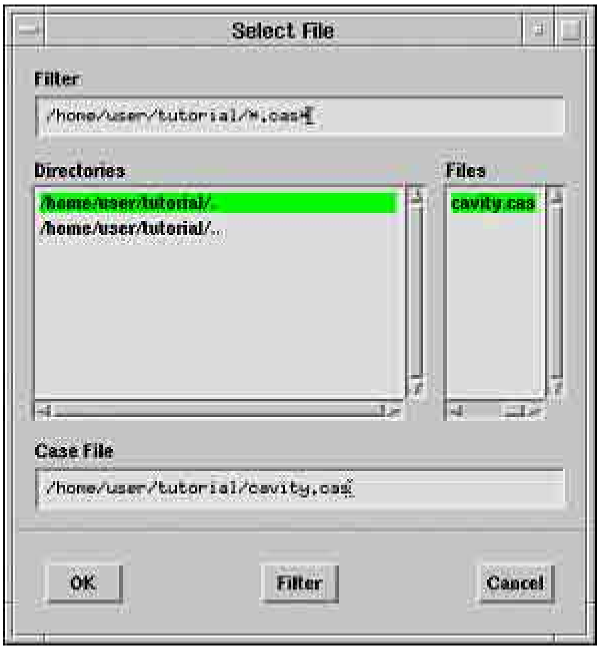


图 1-9 Select（选择文件）窗口

（2）指定文件名。

可以双击 Files（文件）栏中的文件名选定文件，也可以在 Case Files（算例文件）栏中直接输入文件的完整路径和名字以指定文件。

如果是在图 1-10 所示的 XY 文件选择窗口中进行操作，则被选中的文件会被添加到"XY Files"栏中。如果选择的文件不只一个，则所有文件都会列在这个"XY Files"栏中。如果选择有错误，可以在"XY Files"中选中被误选的文件，然后按 Remove（移除）键就可以将它从文件清单中清除。

如果需要保存的文件是算例（case）文件、数据（data）文件或辐射（radiation）文件，用户可以将文件保存为二进制文件，因为二进制文件占用的磁盘空间小，操作速度快。

（3）结束操作。

在上述操作结束后，可以按 OK 按钮，或直接按键盘上的回车键结束操作。

图形显示窗口

图形显示窗口是 FLUENT 软件用来显示图形输出的窗口，见图 1-10。在主控窗口执行菜单操作：

Display –> Options

打开 Display Options（显示选项）面板，在这个面板中可以对图形显示方式进行设定。

执行菜单操作：

Display -> Mouse Button

打开 Mouse Buttons（鼠标按键）面板，在这个面板中可以对图形显示窗口中的鼠标操

作进行设定。

在 WINDOWS 系统中，可以对图形显示内容的打印方式进行设置，方法是点击图形显示窗口左上角的按钮，除了常规的窗口移动、改变尺寸和关闭操作外，WINDOWS 版的 FLUENT 中还提供 copy to clipboard（拷贝到剪切板）、print（打印）和 page setup panel（页面设置）三个命令：

（1）copy to clipboard（拷贝到剪切板）：即将图形窗口中的内容拷贝到 WINDOWS 的剪切板中，以便进行复制、打印等操作。

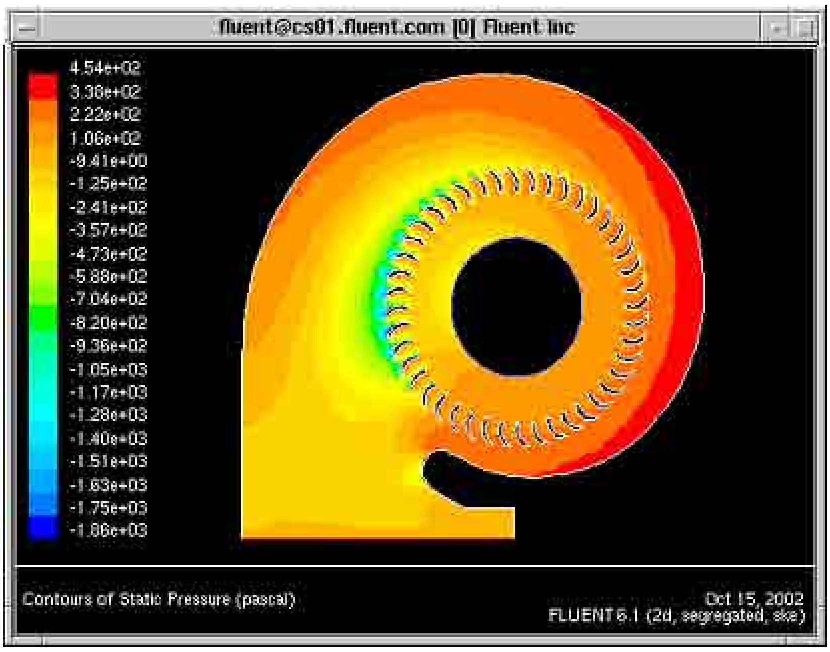


图 1-10 图形显示窗口

（2）print（打印）：即将图形窗口中的显示内容传送到打印机进行打印操作，与打印相关的参数设置在“页面设置”中进行。

（3）page setup panel（页面设置）：即对要打印的图形内容进行图形格式、打印比例等参数的设置，其中图形格式一栏中可以将图形格式设为 Vector（矢量）或 Raster（光栅）。其中 Vector（矢量）格式清晰度高，但操作速度较慢，Raster（光栅）格式清晰度稍差，但操作速度较快，用户可以根据自己的需要决定图形格式。

1.3.2 文字用户界面

FLUENT 除了可以用前面讲的图形用户界面进行操作外，还可以使用类似 DOS 命令的文字命令进行操作。FLUENT 是用 LISP 语言的一个分支 Scheme 语言写成的，如果熟悉这套命令系统的话，用户可以更灵活地操作 FLUENT 系统。

文字菜单系统

在 FLUENT 启动后，用户可以通过在命令提示行中键入命令来完成菜单操作。键入的命令可以被保存、编辑，而且与 Scheme 语言紧密结合，因此通过编程完成复杂的功能操作。文字菜单系统具有树型结构，在 FLUENT 启动后，首先进入的位置就是根目录，在屏幕上用提示符“>”表示根目录位置：



图 1-11 文字菜单系统

键入命令或命令的缩写形式，就可以进入相应的子菜单，比如键入“display”：

> display

就可以进入 display 子菜单，屏幕提示为：

/display>

此时如键入“q”，则返回上级菜单。屏幕显示为：

/display> q

>

命令的缩写形式通常为命令的前几个字母，如果命令是由几个词组成的，则可以将各单词的第一个字母用连字符连在一起，或者直接连在一起作为命令的缩写，比如命令 set-ambient-color，可以缩写成 s-a-c，或者 sac，甚至可以缩写成 sa。这个缩写的基本规律在特殊情况下可能失效，比如命令 light-interpolation 的缩写，不能采用 lint 的形式，因为 li 被包含在单词 light 中，因此上述缩写不被系统看作 l-int 的形式，而是看作 li-nt 形式，li 对应于 light，而 nt 则是系统无法识别的。在这种情况下，可以采用 liin 的缩写形式，或者采用 l-int 的缩写形式，以避免系统错误。

如果键入的命令是以圆括号“(”开始的，则括号内的内容会被完成地传送到系统中，系统会对括号内的内容进行运算，并给出结果。例如：

> (define a 1)

> a
> (+ a 2 3 4)
> 10

上述计算的含义是：

（1） 定义变量 a 等于 1；

（2） 屏幕显示“a”；

（3） 计算“a+2+3+4”的值；

（4） 屏幕显示结果“10”。

文字提示系统

使用文字用户界面还可以完成一些数据、文字的输入工作，比如给变量赋值，在布尔运算中回答 yes（是）或 no（否）等等。这部分工作是在文字提示系统下完成的。所谓文字提示系统是 FLUENT 软件中的一个子系统，这个子系统为用户提供一项在字符界面下的输入功能。

在系统需要用户输入相关参数的时候，在字符界面的提示符后面会出现提示信息。这个提示信息一般由三段组成：第一段是文字信息，提示用户需要输入何种参数信息；第二段被包含在圆括号中，提示用户输入参数的单位；第三段被包含在方括号中，是系统缺省设置的参数数值。其中第二段只在需要标明参数单位的时候出现。

下面是两个提示信息的例子：

filled grids? [no]

shrink-factor [0.1]

如果接受系统的缺省设置，可以直接按回车键；如果不接受系统设置，可以键入一个参数值代替系统设置。

文字提示系统可以输入的数据类型如下：

（1）数字：在数字提示系统中可以输入的数字包括十进制、二进制、八进制和十六进制数，比如数字 31 可以用下列几种方式输入：31，#b11111,#o37 和 #x1f。可见除十进制数字外，其他类型的数字输入都需要在前面加上“#”供系统识别。另外，因为在 Scheme 语言中整数是实数的一个子集，所以在输入上没有区别，就是说输入“2”和“2.0”是完全等价的。

（2）布尔型数据：在需要进行布尔型输入的时候，用户可以直接输入“yes”或“no”，也可以输入缩写形式“y”或“n”，还可以直接输入 Scheme 语言中代表“真”和“假”的布尔型数值：#t（代表“真”）或#f（代表“假”）。

（3）字串：字串的输入需要使用双引号，比如在输入绘图标题等内容时可以输入：“Supersonic Combustion”。在双引号中可以包括任何字符，FLUENT 对此没有限制。

（4）符号：符号的输入不需要使用引号，但是符号之间不允许有空格或逗号。

（5）文件名：文件名实际上就是一个字串，区别是文件名不需要使用引号加以限定，除非文件名中包含空格。在这项规定下，文件名不能被赋值给一个变量，比如：

> (define fn “vla.ps”)

fn

>hc fn

这段语句的原意是将文件名的值“vla.ps”赋给变量 fn，然后在硬盘上写出这个文件，但是在实际执行过程中，系统将把文件内容写入文件“fn”，而不是“vla.ps”，原因就是第一步中的赋值过程没有被系统执行。

（6）列表组（List）：列表项类似于 FORTRAN 语言中的一维数组，其中的各个单元的数值可以逐个输入，也可以整体输入，输入一个空的单元可以结束整个输入过程，比如：

> element(1) [()] 1

> element(2) [()] 10

> element(3) [()] 100

> element(4) [()] ↙

上面四行将 1、10、100 分别输入给列表组的前三个单元，最后一行为空行，按回车键后结束输入。再比如：

> element(1) [(1)] ↙

> element(2) [(10)] ↙

> element(3) [(100)] ↙

> element(4) [()] 1000

> element(5) [()] ↙

可以在原列表组中增加一个单元，同时给它一个值 1000。如果进行下列输入：

> element(1) [(1)] , , ‘(11 12 13) ↙

则前面两个逗号表示原列表组中的前两个单元值保持不变，后面三个单元用整体输入’()的形式增加了三个新的单元数值，最后这个列表组包含了五个单元，其值依次为 1、10、11、12、13。输入（）表示删除当前行及后面所有单元，进行下列操作可以将整个列表组删除：

> element(1) [(1)] () ↙

（7）赋值：除了上面说的文件名输入外，其他输入在被执行前都要按照 Scheme 语言格式被 Scheme 解释器进行赋值操作。比如将一个单位矢量的一个分量设置为 $\frac{1}{3}$，就可以进行下列输入操作，同时系统会完成其中的赋值操作：

/foo > set –xy

x-component [1.0] (/ 1 3)

y-component [0.0] (sqrt (/ 8 9))

上面的输入中方括号中为系统缺省值，圆括号中为输入内容。注意 Scheme 语言的运算格式，（/ 1 3）的意思是 1 除以 3，其结果是 $\frac{1}{3}$。

也可以通过函数定义的方式来达到上述计算目的：

> (define (unit –y x) (sqrt (- 1.0 (∗ x x))))

unit –y

/foo > set –xy
x-component [1.0] (/ 1 3)
y-component [0.0] (unit –y (/ 1 3))

注意上述定义函数的过程中(- 1.0 (* x x)的意思是$1-x^2$，运算符仍然是放在数字前面。在命令执行的过程中按组合键 Ctrl+C 可以随时中断运行。

1.4 FLUENT 文件读入与输出

FLUENT 除了可以读入、输出必要的网格文件、算例文件和进程文件外，还保存了与 FLUENT 公司的其他软件的接口，这些软件包括 FLUENT/UNS、RAMPANT、FIDAP 、FLUENT4 等等，同时还有与 I-DEAS 和 ANSYS 的接口。所有的读入与输出操作均可以在 File 菜单中完成，本小节将逐项进行介绍。

1.4.1 读取网格文件

网格文件是包含各个网格点坐标值和网格连接信息[2]，以及各分块网格的类型和节点数量等信息的文件。在 FLUENT 中，网格文件是算例文件的一个子集，因此在读取网格文件时可以用菜单操作：

File -> Read -> Case

打开菜单并读入网格文件。当然这些网格文件的格式必须是 FLUENT 软件内定的格式。可以用来生成 FLUENT 内定格式网格的网格软件有 GAMBIT、TGrid、GeoMesh 和 preBFC。

除了使用 FLUENT 内定格式的网格文件外，FLUENT 还可以输入其他格式的网格文件。其他格式文件输入的菜单操作是

File –> Import

然后再打开相应格式的输入菜单完成的，其对应关系如下：

（1）GAMBIT：读入 GAMBIT 生成的中性（neutral）网格文件。

（2）preBFC Structured Mesh：读入 preBFC 生成的结构网格文件。

（3）ANSYS：读入 ANSYS 文件。

（4）IDEAS Universal：读入 I-DEAS 通用格式的文件。

（5）NASTRAN：读入 NASTRAN 文件。

（6）PATRAN：读入 PATRAN 分区网格。

（7）CGNS ->Mesh：读入 CGNS（CFD General Notation System，计算流体动力学通用记号系统）[3]格式的网格文件。

---

[2] 即各个网格单元将如何连接在一起的信息。
[3] CGNS 是网格文件的一种格式，据说有可能成为网格文件的工业标准。

（8）CGNS -> Mesh&Data：读入 CGNS 网格文件及相关数据。

需要说明的是，在一个算例文件中既包含了关于网格的信息，也包含了关于边界条件、物性参数以及与计算过程相关的一些参数。在需要改变网格文件时，可以通过读入新文件并将它合并到原来的算例文件中的方式来替换原来的网格信息。不过新的网格必须在结构上和分块方式上与原来的网格相同，否则计算将出现错误。

1.4.2 读写算例文件和数据文件

在 FLUENT 中与数值模拟过程相关的信息保存在算例文件和数据文件里。在保存文件时，可以选择将文件保存为二进制格式或纯文本格式。二进制文件的优点是占用系统资源少，运行速度快。FLUENT 在读取文件时可以自动识别文件格式。FLUENT 还可以根据计算开始前的设置，在间隔一定的迭代步数时自动保存文件。

读写算例文件

如前所述，算例文件中包含了网格信息、边界条件、用户界面、图形环境等信息，其扩展名为.cas，其读入操作可以按下列次序点击菜单，打开文件选择窗口，就可以读入所需的算例文件了：

File -> Read -> Case

与此相类似，按下列次序点击菜单，打开文件选择窗口，就可以保存算例文件了：

File -> Write -> Case

读写数据文件

数据文件记录了流场的所有数据信息，包括每个流场参数在各网格单元内的值以及残差的值，其扩展名为.dat。

数据文件的保存过程与算例文件类似，执行菜单操作：

File -> Read -> Data

打开文件选择窗口就可以读入数据文件。

执行菜单操作：

File -> Write -> Data

则可以保存数据文件。

同时读写算例文件和数据文件

算例文件和数据文件包含了与计算相关的所有信息，因此使用这两种文件就可以开始新的计算。在 FLUENT 中可以同时读入或写出这两种文件，执行菜单操作：

File -> Read -> Case&Data…

打开文件选择窗口，然后选择相关的算例文件完成读入工作，FLUENT 会自动将与算

例有关的数据文件一并读入。类似地，执行菜单操作：

File -> Write -> Case&Data…

打开文件选择窗口，然后选择 Save（保存），就可以将与当前计算相关的算例文件和数据文件同时保存在相应的目录里。

自动保存算例和数据文件

在 FLUENT 中还可以使用自动保存功能以减少人工操作。使用这项功能，可以设定文件保存频率，即每隔一定的迭代步数就自动保存算例和数据文件，菜单操作为：

File -> Write - > Autosave…

于是 Autosave Case/Data（自动保存算例和数据文件）面板被打开，如图 1-12 所示。可以分别设定算例文件和数据文件的保存间隔。在系统缺省设置中，文件保存间隔为 0，即不做自动保存。

在文件名一栏中可以为需要保存的文件命名，如果在命名过程中没有使用扩展名，则系统会自动为所保存的算例文件和数据文件分别加上.cas 或.dat 的后缀。如果在命名过程中使用.gz 或.z 的后缀，则系统会用相应的压缩方式保存算例文件和数据文件。这里.gz 和.z 是 FLUENT 中压缩文件格式。

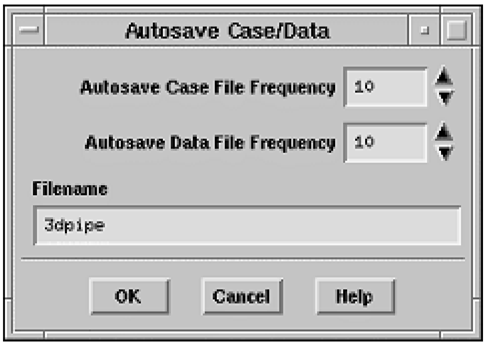


图 1-12 文件自动保存面板

1.4.3 读入 FLUENT/UNS 和 RAMPANT 的算例和数据文件

FLUENT/UNS 和 RAMPANT 是老版本 FLUENT 的两个分支。FLUENT/UNS3、4 两个版本和 RAMPANT2、3、4 三个版本生成的算例文件和数据文件可以直接读入 FLUENT，操作方法与前面讲的 FLUENT 算例文件和数据文件的读入相同，这里不再重复。

使用 FLUENT/UNS 的算例文件系统会自动将求解器设为分离求解器。使用 RAMPANT 的算例文件，系统会自动将求解器设置为耦合显式算法。

1.4.4 读入 FLUENT4 算例文件

执行菜单操作：

File -> Import -> FLUENT 4 Case…

可以打开文件选择窗口，读入 FLUENT4 算例文件。FLUENT4 的算例文件中只包含网格信息和网格分区信息，因此在读入算例文件后，还要进一步设定材料属性、边界条件、求解控制参数等其他相关参数，然后才能开始计算。

1.4.5 读入 FIDAP 算例文件

执行菜单命令：

File -> Import -> FIDAP…

可以打开文件选择窗口，读入 FIDAP 算例文件。与 FLUENT4 相同，FIDAP 算例文件中也只包含网格信息和网格分区信息，在读入算例文件后同样需要设定材料属性、边界条件，然后才能开始计算。

1.4.6 创建与读取进程文件

进程文件（journal file）是一个 FLUENT 的命令集合，其内容用 Scheme 语言写成。可以通过两个途径创建进程文件：一个是在用户进入图形用户界面后，系统自动记录用户的操作和命令输入，自动生成进程文件；另一个是用户使用文本编辑器直接用 Scheme 语言创建进程文件，其工作过程与用 FORTRAN 语言编程类似。

进程文件中可以使用注释语句，Scheme 语言用分号“；”作为注释语句的标志。在一行语句前面使用分号“；”，则表明该行为注释行，用户可以在注释行中为进程文件添加说明信息，也可以锁定一些无用的命令行。

使用进程文件可以重复过去的操作，包括恢复图形界面环境和重复过去的参数设置等等。形象地说，使用进程文件就是重播用户曾经进行的操作，这个重播过程中包含了用户曾经进行过的各种有用和无用的操作过程。因此其使用效率比下面将介绍的描述文件要低。

执行菜单操作：

File -> Write -> Start Journal

系统就开始记录进程文件。此时原来的 Start Journal（开始进程）菜单项变为 Stop Journal（终止进程），点击 Stop Journal（终止进程）菜单项则记录过程停止。

执行菜单操作：

File -> Read -> Journal

打开选择文件窗口，选择要打开的进程文件，然后点击 OK 按钮就可以打开进程文件。

1.4.7 创建记录文件

与进程文件类似，记录文件（transcript file）也是用 Scheme 语言写成的，可以记录用户的所有键盘输入和菜单输入动作，不同的是记录文件不能被读入进行重播操作。记录文件只是为计算做一个完整的操作记录，以便在程序出错时可以回过头来进行检查。

录制进程文件菜单项的下方就是录制记录文件的菜单项，其录制和停止过程也与进程文件类似，即执行菜单操作：

File -> Write -> Start Transcript…

就开始录制记录文件。执行菜单操作：

File -> Write -> Stop Transcript…

则停止录制过程。

1.4.8 读写边界函数分布文件

边界函数分布文件（profile file）用于定义计算边界上的流场条件，比如可以用边界函数分布文件定义管道入口处的速度分布。边界函数分布文件的读写操作如下：

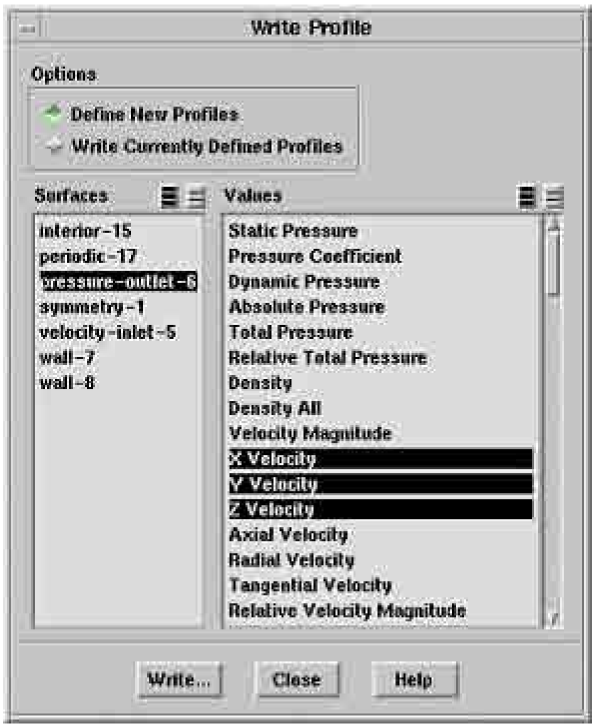


图 1-13 Write Profile（写边界函数分布）面板

（1）执行菜单操作：

File -> Read -> Profile…

打开文件选择窗口，然后选择文件，既可读入边界函数分布文件。

（2）执行菜单操作：

File -> Write -> Profile…

打开定义函数分布文件面板（图 1-13），选择创建新的边界文件还是覆盖原有文件，同时在 Surface（表面）中选择要定义的边界区域，再在 Value（值）中选择要指定的流场参数，点击 Write（写）按钮就可以生成边界函数分布文件。

边界函数分布文件既可以用在原来的算例中，也可以用在新的算例中。例如在管道计算中，用户为出口定义了速度分布，并将它保存在一个边界函数分布文件中。在计算另一个新的算例时，用户可以读入这个文件作为新的管道计算的出口条件。

1.4.9 读写边界条件

如欲将计算中所有边界条件保存在一个文件中，可以执行下列菜单操作：

File -> write-bc

FLUENT 将把边界条件、求解器和计算模型的设置全部保存在一个文件中，文件的格式与算例文件中的 zone（区域）部分相同。

执行下列菜单操作读入边界条件：

File -> Read-bc

读入边界条件后，FLUENT 将把边界条件中的区域名称与计算中使用的区域名称进行对比，名称相同的区域中将采用相同的边界条件。如果边界条件文件中的区域名称在计算模型中没有与之对应的区域，则计算会对这部分边界条件忽略不计。如果计算模型中的区域与边界条件中区域的划分方式有所不同，比如改变了某个区域划分的拓扑结构，那就需要在读入边界条件后，进行边界条件的检查，保证边界条件与其对应的边界是相互吻合的。

如果边界条件文件中的一系列设置是应用到名称相似或名称还没有确定下来的边界上，可以使用星号“*”做省略处理。比如边界条件准备应用到 wall-11、wall-12 和 wall-15 上，就可以通过编辑边界文件将边界名称设置为 wall-*的方式达到目的。

另外，还可以将边界网格写入单独的文件，相应的菜单操作是：

File -> Write -> Boundary Grid

在打开的文件选择窗口中保存文件即可。在用户对网格不满意时，可以先将边界网格保存起来，然后再用 Tgrid 软件读入这个网格文件，并重新生成满意的立体网格。

1.4.10 保存图像文件

图形显示窗口显示的图像可以用很多种方式和文件格式进行保存。保存方式包括使用 FLUENT 软件内部工具进行保存，也可以使用第三方图形软件保存屏幕显示图象。

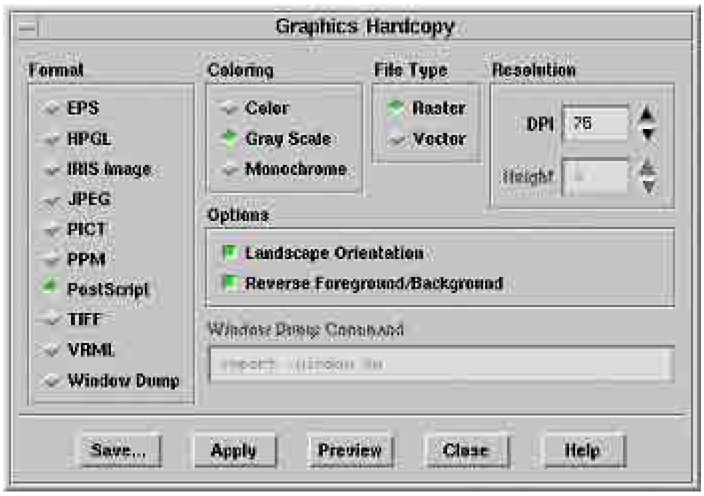


图 1-14 图像文件硬拷贝面板



图 1-15 导出数据面板

FLUENT 内部有一个 Graphics Hardcopy（图形文件硬拷贝）面板，在保存图像文件前，可以使用这个面板对图像文件的保存格式、颜色方案等进行设置，如图 1-14 所示。

在这个面板上可以选择图像文件格式、颜色方案、文件类型、分辨率和方向，并可以预览图像文件。图像文件格式的差别不大，可以根据需要进行选择。颜色方案是选择将文件保存为彩色图像、灰度图像或单色图像。文件类型可以为光栅格式和矢量格式，区别是光栅格式的文件读写速度较快，但是图像质量较差；矢量格式读写速度慢但是图像质量高。

在设置完成后，先用 Preview（预览）按钮检查图像是否满足需要，如果跟自己的预想相去甚远，则重新调整上述几项参数设置；如果对预览结果满意，则可以按 Save（保存）按钮保存图像。如果想了解参数的含义，可以按 Help（帮助）按钮获得在线帮助信息。

1.4.11 导出数据

FLUENT 中设置了与其他软件的数据接口，通过这些接口可以与其他软件进行数据交换，比如 FLUENT 的计算结果和网格数据可以用与其他软件兼容的格式进行保存，从而大大方便了利用其他软件对 FLUENT 的数据结构进行处理。FLUENT 目前可以兼容的导出格式包括 ABAQUS、ANSYS、ASCII、AVS、CGNS、Data Explorer、EnSight (过去称为 MPGS)、FAST、Fieldview、I-DEAS、NASTRAN、PATRAN、RadTherm 和 Tecplot 格式。

需要注意的是，FLUENT 不支持表面（surface）数据。如果导出的文件中带有指定的表面，那么这样的文件将不能重新导入 FLUENT。不过 FLUENT 的网格生成软件 TGrid 支持表面数据。另外，I-DEAS 软件不支持金字塔型的网格划分方式，所以如果网格中带有金字塔型网格的数据，I-DEAS 将无法识别这样的数据。EnSight6 和结构化 FieldView Case+Data 数据可以用 FLUENT 并行版输出。

FLUENT 导出数据是在导出数据面板中完成的，如图 1-15 所示。执行菜单操作：

File -> Export

弹出导出数据面板，执行如下操作就可以将文件导出：

（1）在 file types（文件类型）中选择准备导出的文件类型。

（2）如果选择的是 ABAQUS、ASCII、Data Explorer、I-DEAS Universal、NASTRAN、PATRAN 或 Tecplot，接着在 surfaces（表面）列表中选择要将哪个表面的数据导出。如果不做任何选择，就意味着将整个计算域内的数据导出。对于 RadTherm 文件，只有固壁表面上的数据可以被导出。

（3）除了 ANSYS、FAST Solution 和 RadTherm 三种文件，其他文件格式都要求在 Function to Write 中选择导出变量。

（4）ABAQUS、ASCII、I-DEAS Universal、NASTRAN 和 PATRAN 五种文件格式还要求指定载荷类型，载荷类型包括力、温度和热流通量三种类型。这些载荷将构成有限元计算的载荷矩阵。如上所述，如果没有指定边界面，则整个计算域将被输出到导出文件中，而载荷数据则被写在计算域的外边界壁面上。

（5）ASCII 文件要求指明分隔符和采样点位置，即需要指明上述数据是边界点上的数

据，还是网格单元中心点上的数据。

（6）在需要的时候，还要定义 transient（暂态）导出参数：

1）对于 EnSight 6 和 EnSight Case Gold 文件，可以定义时间步数和文件名称。在计算过程中，当到达定义的时间步数时，所有的几何、速度和标量文件，即所有以.geo、.vel 和.scl1 为扩展名的文件，都会被自动输出到硬盘上。同时时间的值和与其相对应的暂态文件名也被保存在 EnSight 算例文件（即.encas 文件）中。也就是说，暂态文件记录的是某一时刻流场计算的结果，而这些文件的集合则成为流场按时间变化的历史记录。最后，点击 Apply 按钮保存设置。

2）对于其他格式的文件，暂态文件是在执行命令面板上完成的，如图 1-16 所示。命令可以直接键入命令栏中，其一般格式如下：

file/export/ file-type file-name [list-of-surfaces ()] [yes|no] [list-of-scalars q]

其中方括号中的内容是根据文件类型选填的：

- 命令中 file-type（文件类型）是导出文件的类型，比如用 fieldview-uns 代替 file-type，表示导出文件的格式是 Filedview Unstructured。
- file-name（文件名）是导出文件的名称。在文件名称与原有文件名相同时，FLUENT 会弹出提示信息，提示用户是否覆盖原文件。如果用户希望将不同时间步的结果保存到不同的文件中去，可以在文件名中使用字串%t，FLUENT 在导出文件时会将这个字串替换为相应的时间步数。比如，如果用户在命令行中键入的导出文件名为 field-%t，则第一步保存的文件名就是 field-0001，而第一百步保存的文件名就是 field-0100。
- list-of-surfaces（表面列表）。如前面所述，某些文件格式需要指定导出表面。表面列表就是在使用这类格式时，由导出表面的名称或 ID 组成的列表。圆括号“()”是列表的结尾标志。例如，输入“outlet-3 wall-2 5（)”，表示名为 outlet-3、wall-2 的表面和 ID 为 5 的表面将被写入导出文件。注意这里的 ID 不是网格分区的 ID，而是表面的编号。
- list-of-scalars（标量列表）是在使用某些文件格式的时候，指定需要输出的单元函数列表。“q”是这个列表的结尾标志。例如，输入“x-velocity cell-volume q”，将把 x 方向的速度和网格单元的体积作为输出函数写入导出文件。
- yes（是）和 no（否）选项对应的问题是“是否覆盖同名文件？”，即如果同意覆盖硬盘上存在的同名文件，就选 yes；反之，选 no。



图 1-16 执行命令面板

（7）在使用 RadTherm 文件格式时，还要选择写 Heat Transfer Coef.（热交换系数）的方法，即选择 Flux Based（基于通量）形式，还是 Wall Function（壁面函数）形式。

（8）除了 EnSight 格式，其他文件格式都使用 Write...（写出）键保存所有设置。

1.4.12 网格之间解的插值

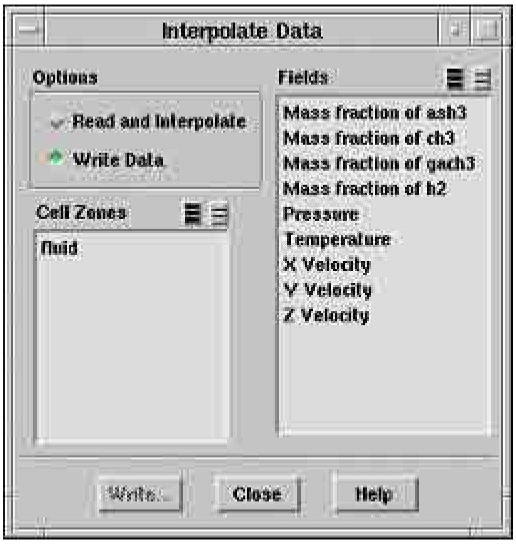


图 1-17 数据插值面板

FLUENT 允许在几何形状确定后，通过插值的方式，在不同网格之间传递数据。比如，可以将六面体网格的计算结果，通过插值转换成混合网格的计算解，然后再利用这个解作

为初始解开始混合网格中的计算。

网格之间的插值计算

网格之间插值计算的步骤如下：

（1）在原始网格上设置计算模型，并计算出一个解。

（2）使用数据插值面板，如图 1-17 所示，将解写入一个插值文件，为在新的网格上进行计算做好准备。

执行下列菜单操作启动数据插值面板：

File -> Interpolate...

在 Options（选项）下选择 Write Data（写数据），然后在 Cell Zones（单元区域）下选择保存数据的区域。如果计算区域中既有流体又有固体，则需将流体数据和固体数据保存在不同的文件上。在 Fields（流场）中选择需要插值的变量，对 FLUENT 中使用的所有变量都可以进行插值处理。最后点击 Write...（写）按钮指定文件名，并保存所有设置。

（3）建立新的算例。读入新的网格，并设置好计算模型、材料性质和边界条件等等。注意一定将原计算中的模型全部引入新的算例，否则将有一些数据在插值过程中被忽略。比如如果原计算中的能量方程模型在新的算例文件中没有被使用的话，则能量数据在插值过程中将被忽略，从而可能导致计算出错。

（4）读入原始计算结果。在图 1-17 的数据插值面板中，在 Options（选项）下选择 Read and Interpolate（读入并插值）选项。在 Cell Zones（网格区域）下选择准备读入和插值的区间。如果所有区域上都没有数据，比如没有进行初始化，或者没有读入插值数据，系统会自动将所有区域选中以保证在插值计算后每个区间上都有数据存在。如果各个区间上已经有了相关的数据，则可以在列出的所有区间中选择几个进行插值运算，并将计算结果赋值给新的网格区域。点击 Read...（读）按钮启动文件选择窗口，选择准备读入的文件，点击 OK 按钮完成读入操作。在计算区间中包含流体和固体两种介质时，上述操作要重复进行两次。

（5）在网格改变后，通量项肯定会出现一定程度上的不平衡，并最终导致流场变量的突变。为了避免或者减轻网格变化带来的不利影响，可以先将亚松弛因子降低，并在新网格中先进行几步运算，使流场完成初始调整。然后再将亚松弛因子调整到原来的值，正式开始在新网格上的计算。

插值文件的格式如下：

- 第一行是插值文件的版本，对于 FLUENT 5.0 这个数是 1.0，对于 FLUENT 6.0 这个数是 2.0。
- 第二行是计算的空间维数（二维或三维）。
- 第三行是总的网格点数。
- 第四行是总的变量数量（压强、温度、速度等等）。
- 从第五行开始是变量名列表。通过菜单命令[Display]-[Contours]可以看到所有变量的完整列表。

- 在变量列表的后面是坐标 x、y 的值，在三维问题中还包括 z 的值。
- 所有变量的数值按照第五行的排列次序排列在坐标值的后面，并且变量的排列次序与坐标的排列次序是相同的。二者的数量应该与第三行给出的值相符。下面是一个例子：

```
2
2
34800
3
x-velocity
pressure
y-velocity
-0.068062
-0.0680413
...
```

1.4.13 读入 Scheme 源文件

Scheme 语言的源文件可以用三种方式读入，一种是在菜单中作为 Scheme 文件读入，第二种是在菜单中作为进程文件读入，第三种是用 Scheme 语言的函数命令读入。

如果 Scheme 文件比较大，可以通过下列菜单操作读入：

File -> Read -> Scheme...

或者使用函数命令读入：

> (load "file.scm")

小的 Scheme 文件可以用下列菜单操作读入：

File -> Read -> Journal...

或者用命令 file/read-journal 读入，还可以用“.”或 source 命令读入：

> . file.scm

或

> source file.scm

均可以达到上述目的。

1.14.14 .fluent 文件

在 FLUENT 启动过程中，系统会在用户的安装目录中寻找一个叫.fluent 的文件。如果找到这个文件，FLUENT 将用 Scheme 的载入函数将这个文件读入系统。这个文件中可以包含 Scheme 语言写成的用于定制 FLUENT 运行的函数命令。

1.14.15 保存外观设置

使用下列菜单命令可以保存当前窗口、面板等的设置：

File -> Save Layout

如果用户希望保持当前使用的窗口、面板的设置形式，可以使用这个菜单命令将当前设置保存在根目录下的文件 cxlayout.scm 中。

1.14.16 算例和数据文件的格式

FLUENT 的算例和数据文件通常被分成几段（section），每个数据段都用括号括起来，每个数据段的开头都用一个整数作为标志，标志其数据类型。这种分段方式便于数据的检索、分类和更新。数据类型标志的后面是用括号单独括起来的、关于数据内容的头信息，数据内容则用独立的括号包围。

1.5 FLUENT 中的单位制

只要选择正确的转换因子，就可以在 FLUENT 中混合使用不同的单位制。比如在计算热力学问题时，使用英制的瓦特作功率单位，同时却可以使用国际单位制的米作为长度单位。在 FLUENT 的内核中是以国际单位制进行计算的，所谓混合单位实际上是在输入输出过程中照顾用户习惯的一种方法，最后所有单位制还是要在 FLUENT 内部统一为国际单位制，因此对计算过程没有影响。

1.5.1 对单位的限制

在 FLUENT 中对下述问题只能使用国际单位制进行输入：

（1）边界函数分布文件。

（2）源项。

（3）自定义场变量。

（4）由外部绘图软件生成的数据。

（5）用户自定义函数（UDF）。

如果在计算过程中使用温度的多项式定义材料性质，则温度的单位必须是开氏温度或兰氏温度。如果采用摄氏温度和开氏温度作为温度单位，则在使用温度多项式进行计算时，多项式的系数必须采用开氏温度；如果使用兰氏温度和华氏温度，则多项式的系数必须采用兰氏温度。

1.5.2 网格数据的单位

有些网格生成软件允许采用多种单位制作为网格长度的单位，但是 FLUENT 中只允许使用国际单位制，因此在使用那些不是用国际单位制定义的网格数据时，用户需要用后面章节中介绍的方法进行单位制的转换。

1.5.3 FLUENT 内建的单位制

FLUENT 有四种内建的单位制：英制、国际单位制、厘米-克-秒制和“缺省”单位制。在图 1-18 所示的单位设定面板中，使用 Set All To（将全部数据转换为……）下面的按钮可以将各种单位转换为统一的单位。执行下列菜单操作弹出单位设定面板：

File -> Units

转换单位制的步骤如下：

（1）在 Quantities（数量）下选定要转换单位的变量。

（2）在 Units（单位）下将显示当前使用的单位，再在 Set All To（将全部数据转换为……）下点击一个按钮，则原来的数据将转换为按钮所对应的单位制。在进行这一步的操作时，可以见到反白显示的单位从原来的位置跳到新的位置。

比如将面积的单位从国际单位制的平方米转换为英制的平方英尺，就可以先选择变量 area（面积），此时 Units 下面反白显示当前的单位是 m^2（平方米），再点击 british（英制）按钮，则可以看见反白的显示条从原来的地方跳到了 ft^2（平方英尺）的位置。

FLUENT 中的“缺省”单位制与国际单位制的唯一区别是角度的单位是“度”，而不是“弧度”。



图 1-18 单位设定面板

1.5.4 调整数据单位

如果用户不喜欢使用 FLUENT 提供的四种内建单位制，可以使用图 1-18 的单位设定面板对需要改变的变量进行选择，甚至通过给定转换因子设定新的单位。

列出现有单位

点击单位设定面板下面的 List（列表）按钮可以将系统内部所有的变量和单位以列表的形式显示在窗口中，同时显示各种单位的转换因子（Factor）和偏离值（Offset）。

改变一个变量的单位

与内建单位制的设定方法相类似，选定一个变量后，除了通过点击 Set All To（将全部数据转换为……）下面的按钮的方式将单位转换到另一个单位制外，还可以直接在 Units（单位）列表中选择一个单位作为变量的单位。

定义新单位

点击 New...（新）按钮可以打开 Define Unit（定义单位）面板定义新的单位。如图 1-19 所示，如果想为时间变量定义一个新的变量 hr（小时），可以先在 Quantities 下面选择变量 time，然后点击 New...（新）按钮进入 Define Unit（定义单位）面板，指定 Unit（单位）为 hr（小时）、Factor（转换因子）为 3600、Offset（偏离值）为 0，按 OK 键结束。

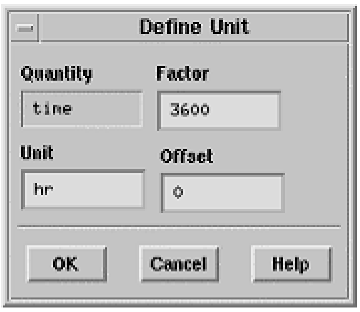


图 1-19 定义单位面板

转换因子的含义是新的单位与国际单位制之间的比例系数。在这个例子中，时间的新单位 hr（小时）是国际单位制 s（秒）的 3600 倍，即 1 小时等于 3600 秒，所以转换因子等于 3600。

1.6 FLUENT 的计算策略

任何工作都需要有一个计划，用 FLUENT 进行计算也不例外，计算前要在头脑中先勾勒出一个计划，然后再按照计划进行工作。

1.6.1 如何制订计划？

制订计划之前，需要了解下列问题：

（1）确定工作目标：即明确计算的内容是什么？计算结果的精度应该有多高？

（2）选择计算模型：要考虑如何划定流场？流场的起止点在哪里？边界条件怎么定义？是否可以用二维进行计算？网格的拓扑结构应该是什么样的？

（3）选择物理模型：流动是无粘流、层流，还是湍流？流动是可压的，还是不可压的？需要考虑传热问题吗？流场是定常的，还是非定常的？在计算中是否还要其它的物理问题？

（4）确定求解流程：要计算的问题能否用系统缺省的设置简单地完成？是否有什么窍门可以加快计算的收敛？计算机的内存是否够用？计算需要多长时间？

仔细思考这些问题可以更好地完成计算，否则在计算的过程中就会经常遇到意想不到的问题，并且经常返工，浪费时间，降低工作效率。

1.4.2 计算步骤

工作计划确定下来后，就可以按照下面的基本步骤开始计算：

（1）定义流场的几何参数并进行网格划分。

（2）启动相关的求解器。

（3）输入网格。

（4）检查网格。

（5）选择求解器格式。

（6）选择求解所用的基本方程：层流还是湍流？有没有化学反应？是否考虑传热？是否需要其它的物理模型，比如是否使用多孔介质模型？是否使用风扇模型？是否使用换热器模型？

（7）定义物质属性。

（8）定义边界条件。

（9）调整解的控制参数。

（10）初始化流场。

（11）开始求解。

表 1-1 FLUENT 计算步骤及对应菜单项

求解步骤	对应菜单项
3. 输入网格	File
4. 检查网格	Grid
5. 选择求解格式	Define
6. 选择基本方程	Define
7. 物质属性	Define
8. 边界条件	Define
9. 调整求解控制参数	Solve
10.初始化流场	Solve
11.计算求解	Solve
12.检查结果	Display 或 Plot 或 Report
13.保存结果	File
14.根据结果对网格做适应性调整	Adapt

（12）计算结束后检查计算结果。

（13）保存结果。

（14）如果结果不理想，可以考虑调整网格或者物理模型重新进行计算。

在网格划分阶段，可以用 GAMBIT、TGRID 或其它 CAD/CAE 软件进行网格划分，具体的操作方法可以参考本书第二篇关于预处理技术的介绍。

1.7 FLUENT 的计算方式

FLUENT 提供三种计算方式，即分离方式、耦合隐式和耦合显式。这三种计算方式都可以给出精确的计算结果，只是针对某些特殊问题时，某种计算方式可能比其它两种方式更快一些。

分离计算和耦合计算的区别在于求解连续、动量、能量和组元方程的方法有所不同。分离方式是分别求解上面的几个方程，最后得到全部方程的解，耦合方式则是用求解方程组的方式，同时进行计算并最后获得方程的解。两种计算方式的共同点是，在求解附带的标量方程时，比如计算湍流模型或辐射换热时，都是采用单独求解的方式，就是先求解控制方程，再求解湍流模型方程或辐射方程。显式和隐式的区别在于对方程的线化方式有所不同。后续章节中还会详细介绍这三种计算方式。

分离方式一般用于不可压流或弱可压流的计算。耦合方式则通常用于高速可压流计算。而在 FLUENT 中，两种方式都可以用于可压和不可压流动计算，只是在计算高速可压流时，耦合方式的计算结果更好一些。

FLUENT 求解器的缺省计算方法是分离算法，但是对于高速可压流、彻体力强耦合型问题（比如浮力问题或旋转流动问题）、超细网格计算问题等类型的问题，最好还是使用耦

合隐式计算方式。这个求解器收敛速度更快，只是需要占用更大的内存。

上面提到的几个问题虽然最好采用耦合隐式方式进行计算，但是在内存不够时，也可以考虑使用分离或耦合显式方法进行计算。耦合显式计算也是将能量方程与其它方程耦合在一起进行计算，但是所需内存更小，而计算时间则比较长。

需要注意的是分离求解器中使用的几个物理模型在耦合求解器中是不能使用的，这些模型包括：

（1）多相流模型。

（2）混合浓度/PDF 燃烧模型。

（3）预混燃烧模型。

（4）污染物构成模型。

（5）相变模型。

（6）Rosseland 辐射模型。

（7）特定质量流周期流模型。

（8）流向周期性换热模型。

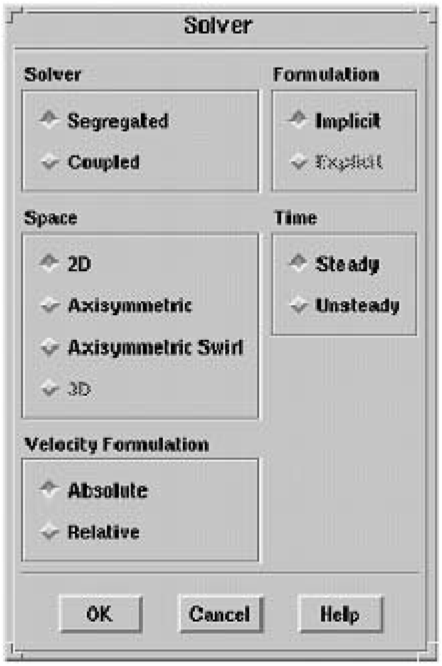


图 1-20 在 Solver（求解器）面板中选择求解方式

计算方法可以通过 Solver（求解器）面板进行选择。Solver（求解器）面板的启动方法为：

Define -> Models - Solver…

在缺省设置下，系统选用的是分离求解器。要选择耦合隐式算法，就在 Solver（求解器）下选择 Coupled（耦合），在 Formulation（算法）下选择 Implicit（隐式）。如要选择耦

合显式方式，则在 Solver（求解器）下选择 Coupled（耦合），在 Formulation（算法）下选择 Explicit（显式）。

1.8 例题：方腔流动计算

下面利用 FLUENT 光盘上的一个网格文件来做个简单的计算。

我们要算的问题如图 1-21 所示，在一个菱形的二维空腔中充满等密度的空气，菱形每边长为 0.1m，顶角为 60º，其顶板以 0.1m/s 的速度向右移动，同时带动空腔内流体的流动。流场的雷诺数大约为 500，为层流流动。我们要算的就是空腔内的流场，也就是说通过计算我们要确定空腔内流体的流动参数分布。

1.8.1 计算过程概述

这个问题是个简单的二维流动计算问题，流动是层流，没有热传导，也没有任何其它的物理模型需要考虑。与问题相关的网格划分问题我们暂时不用考虑，光盘上有现成的网格文件，将它读入 FLUENT 就可以了。

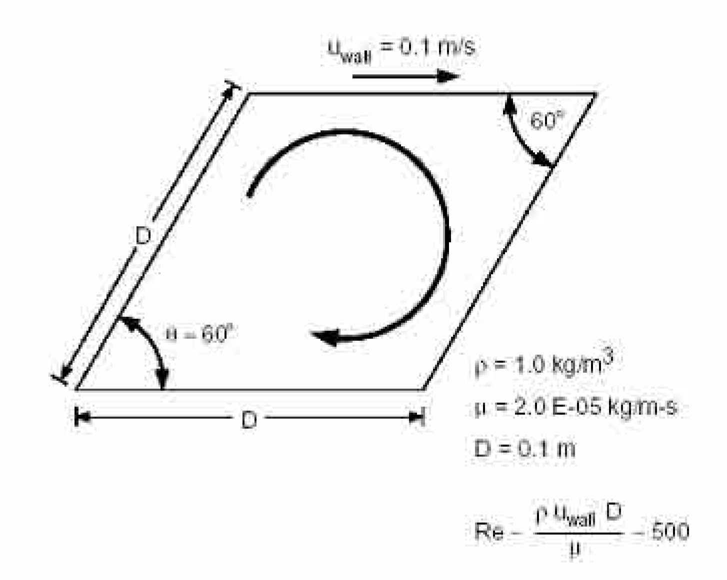


图 1-21 顶板移动孔穴中的流动

计算步骤因此可以简化为：

（1）读入网格文件并进行检查。

（2）选择缺省的分离计算方式。

（3）定义物理模型。
（4）指定流体物质属性。
（5）保存问题设置。
（6）初始化。
（7）开始计算。
（8）保存结果。
（9）检查结果。
下面是具体操作过程：

1.8.2 读入网格文件并进行检查

启动 FLUENT 的二维计算版本。进行下列菜单操作就会出现图 1-22 所示的文件选择对话框：

File -> Read -> Case…

读入网格

在本算例中要使用的网格文件已经被保存为 FLUENT 格式，因此可以直接将它读入。

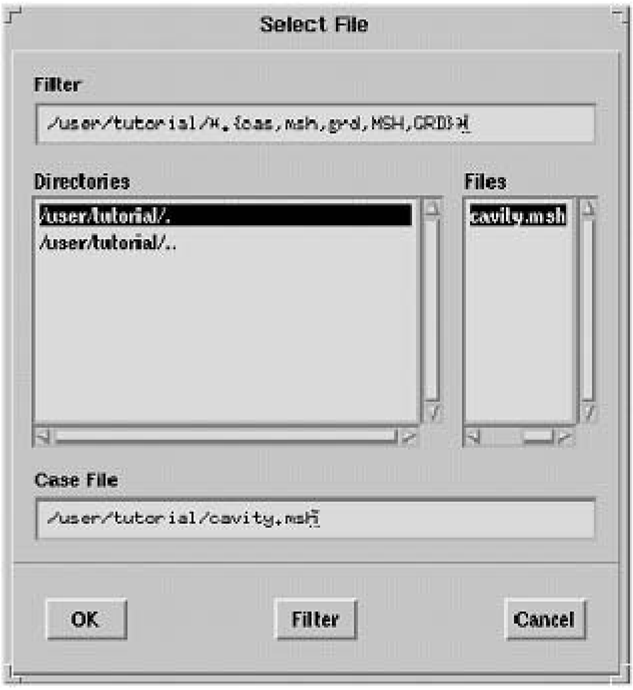


图 1-22 文件选择对话框

检查网格

网格读入成功后，需要对它进行检查。检查的目的是看网格文件是否可以用于计算。执行菜单操作：

Grid -> Check

在控制台窗口中就可以看到网格覆盖范围、体积统计数据、连通性信息等数据，如图 1-23 所示。

比较容易发现的网格错误是网格体积出现负值。如果网格的体积出现负数，就需要重新进行网格划分，以便去掉这种在物理上不真实的网格形式。要显示体积为负数的网格，执行菜单操作：

Adapt -> Iso-Value…

选择显示体积数值小于零的体积。

显示网格

```
Grid Check

 Domain Extents:
   x-coordinate: min (m) = 0.000000e+00, max (m) = 1.500000e-01
   y-coordinate: min (m) = 0.000000e+00, max (m) = 8.660000e-02
 Volume statistics:
   minimum volume (m3): 7.156040e-05
   maximum volume (m3): 7.157349e-05
     total volume (m3): 8.660000e-03
 Face area statistics:
   minimum face area (m2): 9.089851e-03
   maximum face area (m2): 9.091221e-03
 Checking number of nodes per cell.
 Checking number of faces per cell.
 Checking thread pointers.
 Checking number of cells per face.
 Checking face cells.
 Checking face handedness.
 Checking element type consistency.
 Checking boundary types:
 Checking face pairs.
 Checking periodic boundaries.
 Checking node count.
 Checking nosolve cell count.
 Checking nosolve face count.
 Checking face children.
 Checking cell children.
Done.
```

图 1-23 网格检查结果

在网格检查工作结束后，可以显示网格，这样就可以直接看到网格划分的具体模样，

并通过经验对网格划分的质量做出判断。显示网格的菜单操作为：

Display -> Grid…

在弹出的网格显示面板上点击 Display（显示）按钮将启动一个图形显示窗口，并在窗口中开始绘制网格。可以看到如图 1-24 所示的网格图形。点击 Close（关闭）按钮关闭显示面板。

在显示网格的窗口中，按住鼠标中键并拖动，可以将选中区域放大显示。这个功能可以用于显示网格的局部，便于看清网格细节。

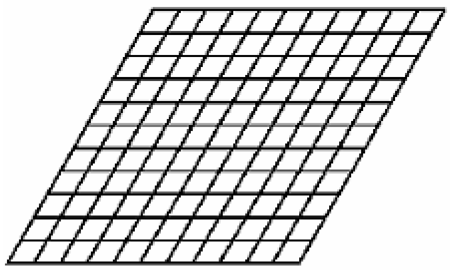


图 1-24 网格划分显示

1.8.3 选择求解方式

在这个算例中，因为流动速度很低，所以可以认为流动是不可压流动，也就是说可以使用 FLUENT 缺省的分离求解器进行计算，因此不必改变任何设置就可以进行计算了。当然，如果要尝试使用另外两种耦合方式进行计算的话，也可以执行下列菜单操作启动 Solver（求解器）面板并选择计算方式：

Define -> Model

1.8.4 定义物理模型

FLUENT 中的缺省设置是层流计算。因为这个算例中的流动是层流，所以不需要改变任何东西。如果需要改变设置，使用其他模型的话，可以执行下列菜单操作：

Define -> Model-> Viscous...

打开 Viscous（粘性模型）面板并进行修改。

1.8.5 定义流体材料性质

执行下列菜单操作启动 Materials（材料）面板，如图 1-25 所示：

Define -> Materials…

如果不想用空气作为计算对象，可以从材料数据库中选择其他的材料作为计算对象，或者可以创建一种新的材料做计算对象。在本算例中，只要选择 air（空气），再对空气性质进行一些调整就可以了。这些修改是：

（1）将 Density（密度，单位为 kg/m^3）的值改为 1.0。

（2）将 Viscosity（粘度，单位为 kg/m-s）的值改为 2e-05。

（3）点击 Change/Create（改变/创建）按钮保存新的数值，然后关闭面板，结束物质属性设置工作。

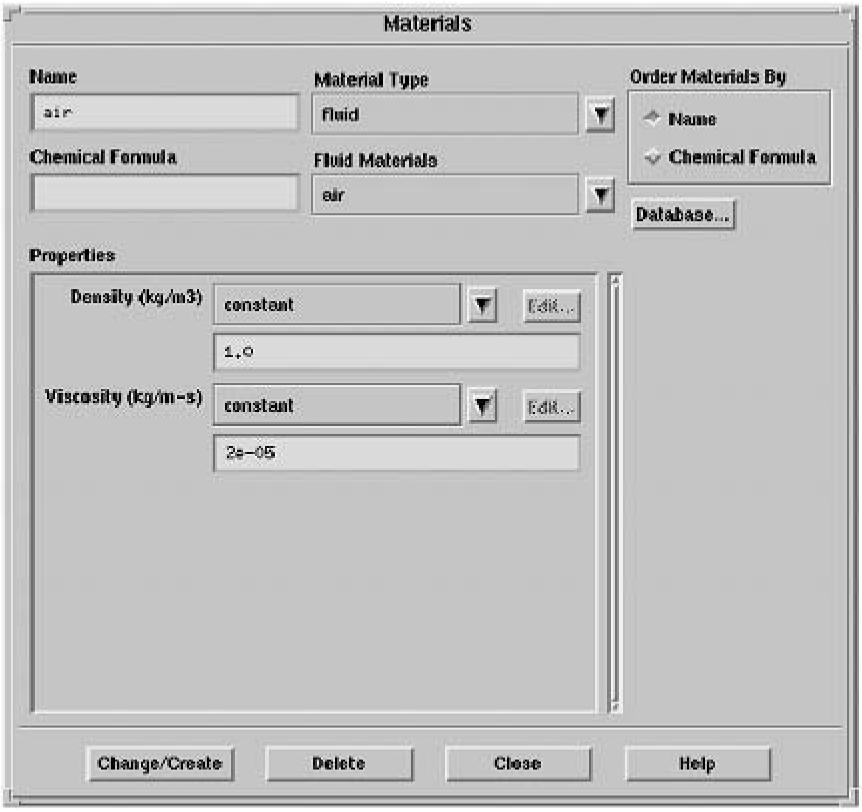


图 1-25 Material（材料）面板

1.8.6 定义边界条件

边界条件的设定是在边界条件面板中完成的。执行下列菜单操作启动这个面板，如图 1-26 所示：

Define -> Boundary Conditions…

在这个面板上可以改变边界类型，并设定边界条件参数。

为指定区域设置边界条件的操作方法如下：

（1）在 Zone（区域）栏中选择边界区域。

（2）在 Type（类型）栏中选择边界类型。

（3）点击 Set…（设置）按钮设置具体数值，也可以在选择了边界类型后，直接双击 Zone（区域）栏中的边界区域名来设置具体数值。

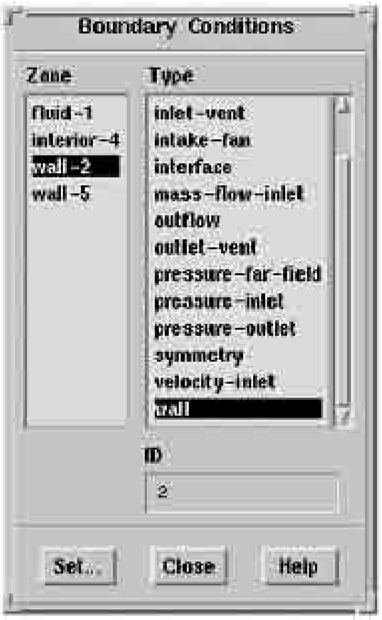


图 1-26 边界条件面板

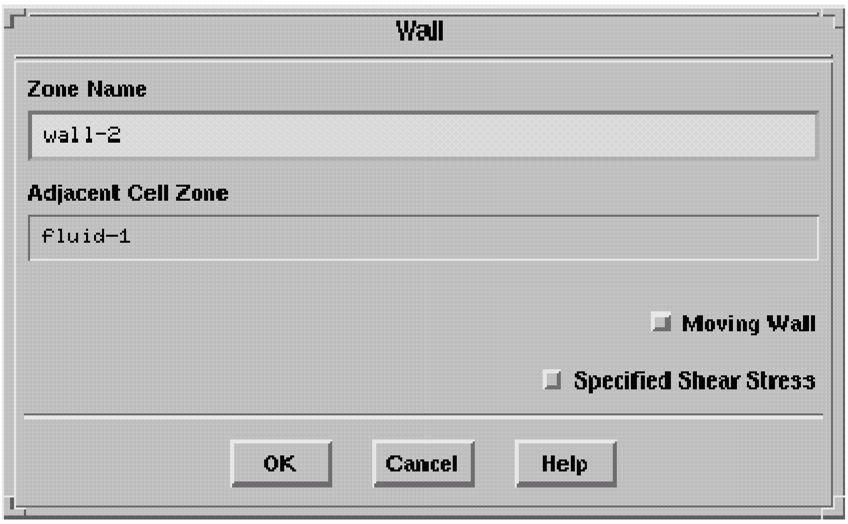


图 1-27 壁面设置面板

在本算例中，需要将顶板速度设置为 0.1m/s 来模拟空腔上方移动的盖板。如果不知道 Zone（区域）栏中哪个边界是需要被设置为移动顶板，可以在显示网格的图形窗口中用右

键点击代表顶板的那条边，这样边界条件面板中就会自动选中相应的边界。在这个例子中是 wall-2，然后再点击 Set…（设置）按钮就可以进行设置了，如图 1-27 所示。在进行上述操作的同时，与边界 wall-2 相关的信息会显示在控制台窗口中。

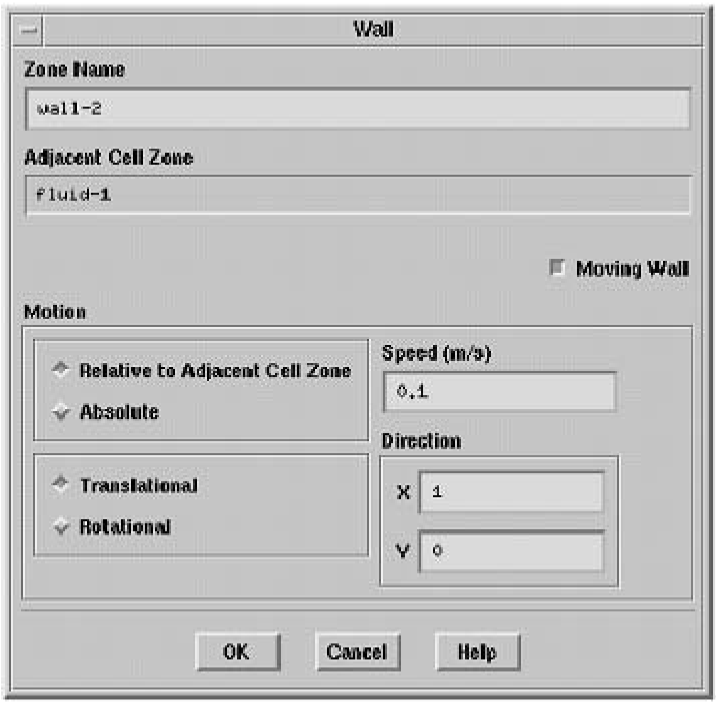


图 1-28 移动壁面设置面板

要设置移动壁面条件，必须选择 Moving Wall（移动壁面）选项，操作的结果是壁面设置面板将展开成图 1-28 的样子。在缺省情况下，系统假定壁面是平动的，运动方向与 X 轴方向相同，因此在本算例中只要将速度设为 0.1m/s 就可以了。

在旋转型壁面条件设置时不必在选择相对运动还是绝对运动上过多犹疑，因为周围的流体是不动的，所以上述两项选择是等效的。

在输入完速度值后就可以点击 OK 按钮保存结果并关闭壁面设置面板了。本算例中的其它三个壁面都是静止壁面，这与系统的缺省设置相同，所以不必再做什么设置，直接关闭边界条件设置面板就可以了。

1.8.7 调整计算控制参数

执行下列菜单操作可以改变亚松弛因子、多重网格参数和其它计算控制参数：

Solve -> Controls

在本算例中不用改变这些设置。

绘制残差曲线

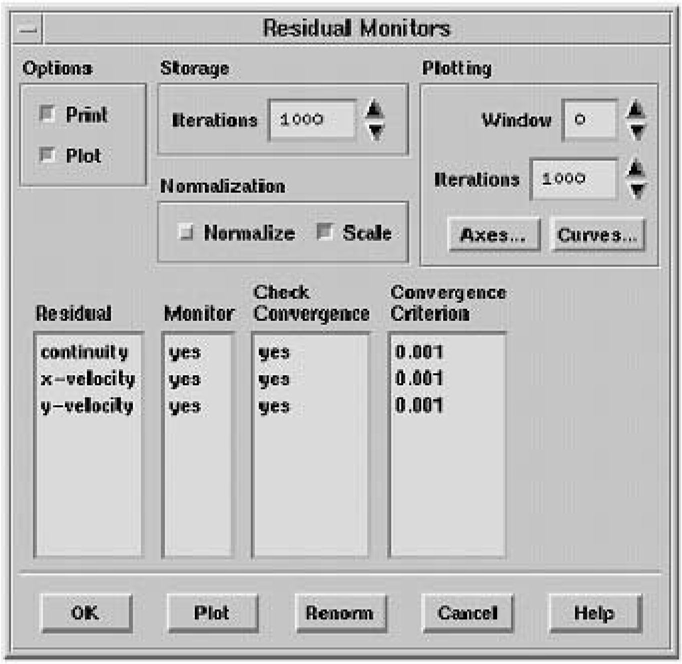


图 1-29 残差监视面板

现在设置工作基本完毕，可以打开一个图形窗口显示计算的残差曲线，监视计算过程。执行下列菜单操作打开 Residual Moniters（残差监视器）面板，如图 1-29 所示：

Solve -> Monitors -> Residual…

在 Options（选项）下选择 Plot（绘制），则在计算进行过程中会打开残差监视窗口。因为在初始化之前各变量尚未赋值，所以显示残差需要在初始化后进行。

残差值越来越小，表示计算结果正在向一个极限收敛。如果残差越来越大，则说明计算已经发散。

1.8.8 保存算例文件

前面所设置的参数均被保存在算例文件中，一般在进行计算前都要进行一次保存。计算的数值结果则保存在数据文件中，计算结束时必须保存以免数据丢失。

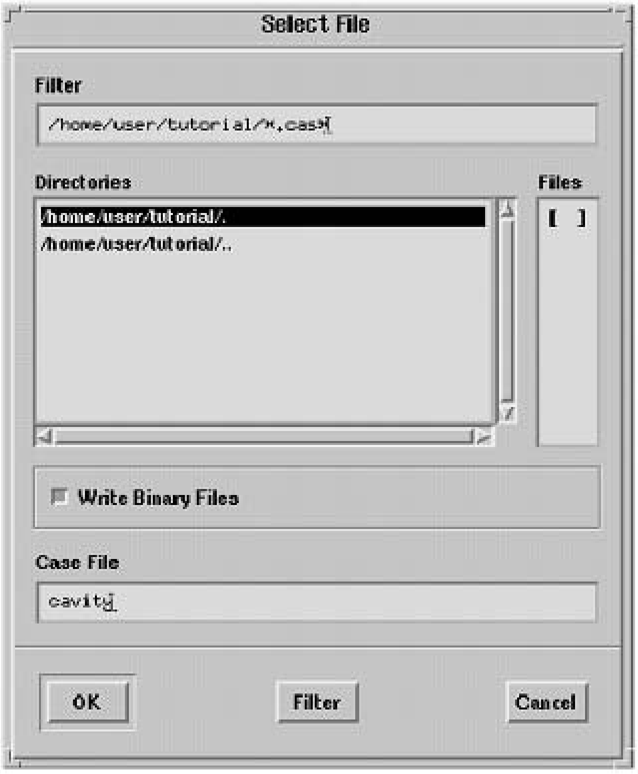


图 1-30 保存算例文件

1.8.9 开始计算

初始化流场

在开始迭代前，必须先初始化流场。所谓初始化，其实就是给各流场参数赋初始值，以便迭代计算可以有一个起点。初始化流场的方式有两种，一种是初始化流场边界，另一种是直接初始化流场变量。执行下列菜单操作打开 Soluton Initialization（求解初始化）面板，如图 1-31 所示：

Solve -> Initialize -> Initialize…

在本算例中，空腔流动必然发展成为旋转流动，因此可以将速度值设为 0，也就是保留缺省设置，然后点击 Init（初始化）按钮开始初始化计算，最后点击 Close（关闭）按钮关闭面板。

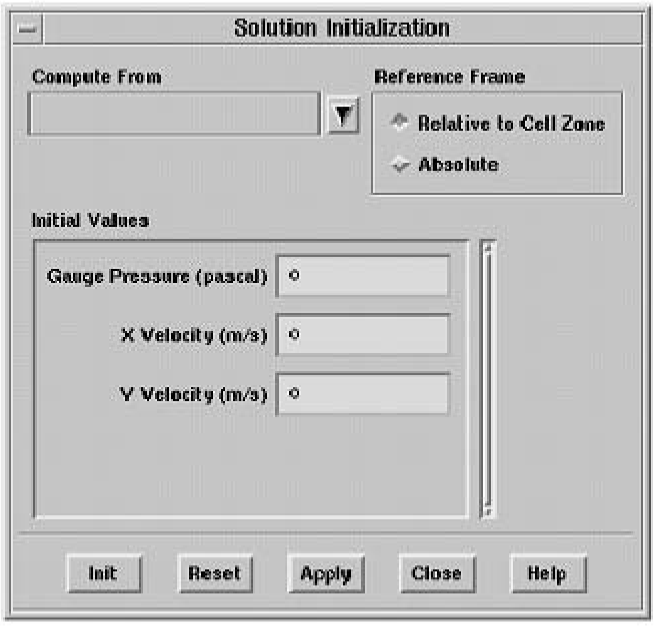


图 1-31 初始化面板

开始计算

现在可以开始迭代计算了。执行下列菜单操作打开迭代面板，如图 1-32 所示：
Solve -> Iterate…

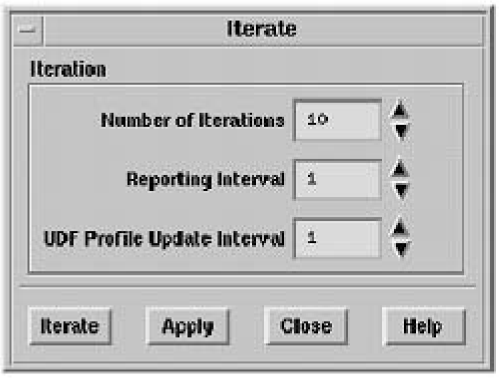


图 1-32 迭代面板

在 Number of Iterations（迭代次数）中填入 10，然后点击 Iterate（迭代）按钮开始计算。在 10 次迭代后，从图形窗口中可以看到图 1-33 所示的残差曲线。残差曲线下降表明计算在收敛。

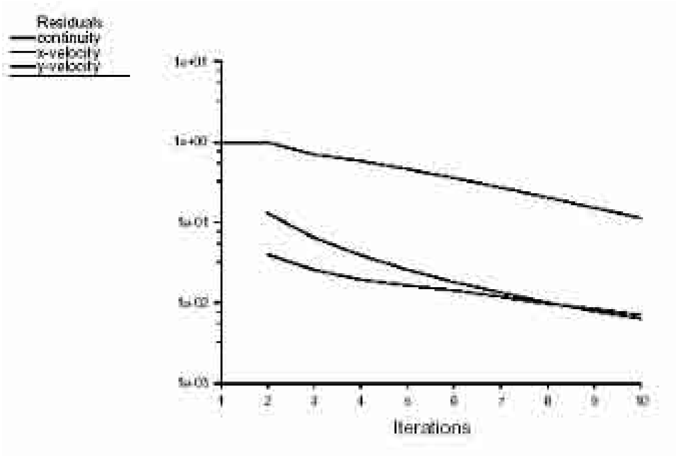


图 1-33 残差曲线

在计算过程中可能还想看看流场的变化过程。执行下列菜单操作打开一个速度矢量面板，如图 1-34 所示：

Display -> Velocity Vectors…

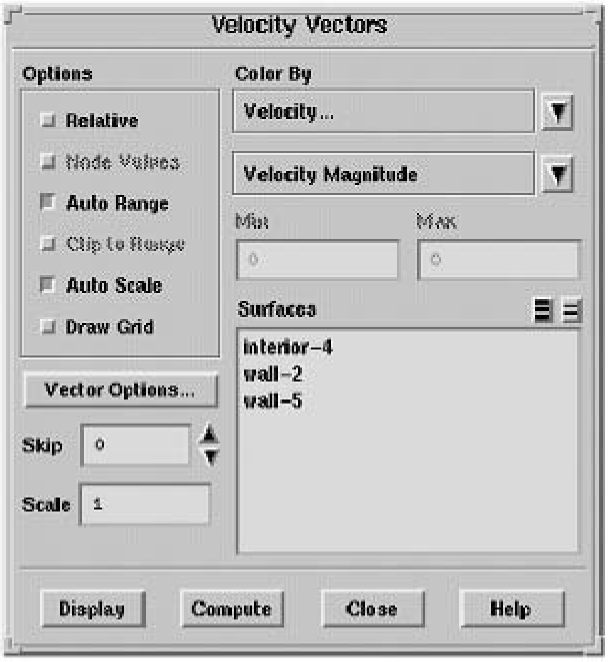


图 1-34 速度矢量面板

这个面板中的缺省设置将绘制一个速度矢量图，其色彩将随速度大小发生变化。点击 Display（显示）按钮，可以看到一个如图 1-35 所示的速度矢量图。

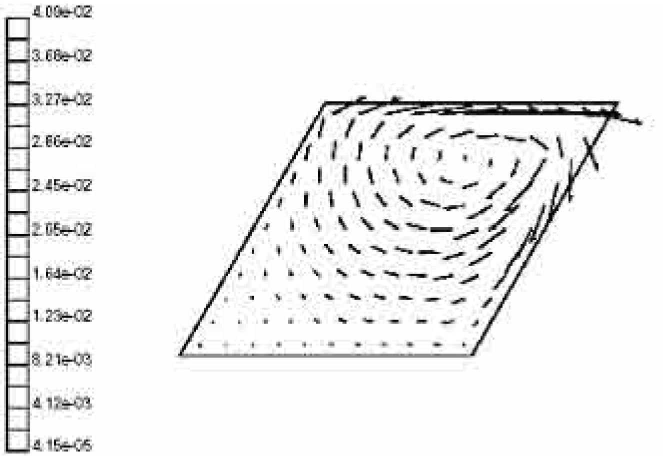


图 1-35 速度矢量图

从图中可见，虽然只计算了 10 步，空腔内流体已经明显开始旋转。这表明计算在按照预想的进程发展，因此可以在迭代面板中将迭代次数重新设为 100，然后开始计算。事实上在计算到 50 步时，就基本达到了系统缺省的收敛条件。此时残差曲线将如图 1-36 所示。

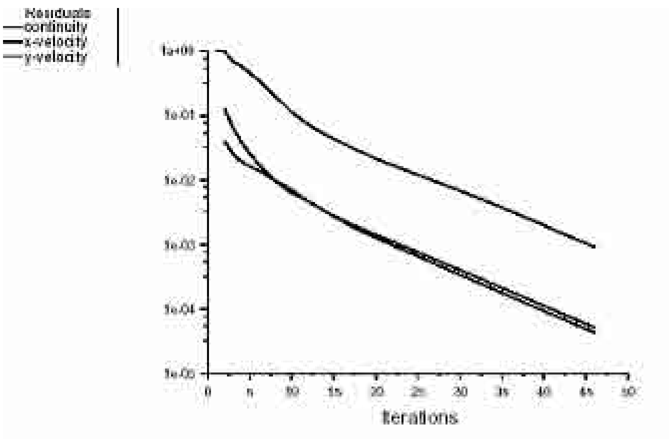


图 1-36 迭代 50 步后的残差曲线

现在可以检查收敛后的计算结果，并将结果保存起来。

1.8.10 保存计算结果

选择同时保存算例文件和数据文件的方式，并将文件命名为 cavity，则输入数据和计算结果被分别保存在算例文件 cavity.cas 和数据文件 cavity.dat 中。在将来可以打开这两个文件继续进行计算或进行后处理操作。在成功保存文件后，可以重新调入这些文件进行计算。

1.8.11 检验计算结果

前面已经绘制过速度矢量线，在 FLUENT 中还可以绘制等值线。

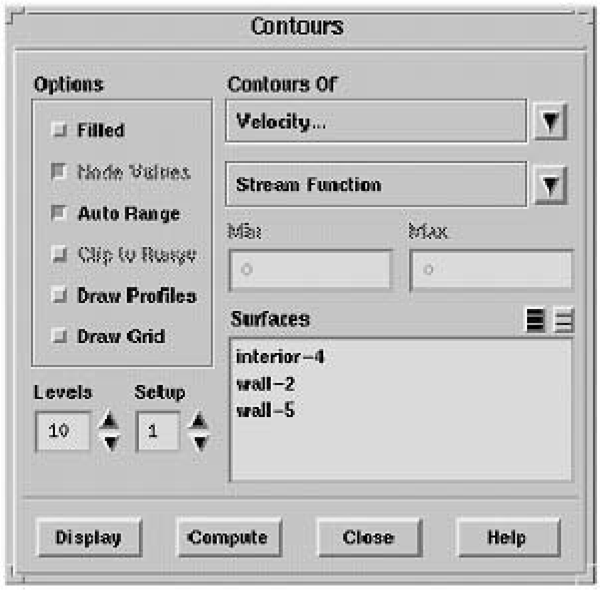


图 1-37 等值线面板

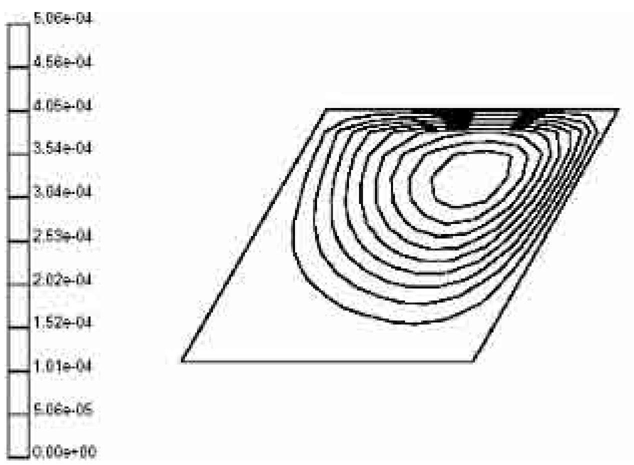


图 1-38 流函数等值线

执行下列菜单操作打开一个如图 1-37 所示的等值线面板：

Display -> Contour…。

在 Contours Of 的下拉菜单中选择 Velocity…(速度)，在下面一个下拉菜单中选择 Stream Function（流函数），再将左下角的 Levels 设为 10，点击 Display（显示）按钮，则系统

会绘制一个如图 1-38 所示的等值线图。在等值线显示完毕后，点击 Close（关闭）按钮关闭等值线面板。

1.8.12 退出 FLUENT

看完计算结果后就可以退出了，在菜单上进行下列操作退出：

File -> Exit…

这个简单的例子主要是帮助用户理解怎样使用 FLUENT。在第二章中我们将增加难度，计算包括一个稍微复杂一些的例子，进一步深化求解过程的讲解。

1.9 本章小结

在这一章中给大家介绍了 FLUENT 软件的基本情况，包括与 FLUENT 同时发布的几个软件及其主要功能，FLUENT 软件及与它配合使用的软件的安装和启动过程和系统要求，FLUENT 的主要计算方式和适用范围，FLUENT 的图形用户界面和文字用户界面，FLUENT 与其他 CAD/CAE 软件的接口，各种文件在 FLUENT 中的导入和导出方法，FLUENT 的计算规划和计算步骤。最后通过一个简单的算例展示了用 FLUENT 软件进行计算的基本流程。

第二章 FLUENT 的计算步骤

本章通过一个稍微复杂一些的算例再次演示 FLUENT 的求解过程。这个算例的内容是计算一个二维弯管中的湍流流动和热传导过程，在这个算例中可以看到 FLUENT 计算的标准流程，其中包括：

（1）如何读入网格文件。

（2）如何使用混合的单位制定义几何模型和物质属性。

（3）如何设定边界条件和和物质属性。

（4）如何初始化计算并用残差曲线监视计算进程。

（5）如何用分离求解器计算流场。

（6）如何用 FLUENT 的图形显示功能检查流场。

（7）如何用二阶精度离散格式获得更高精度的流场。

（8）以温度梯度为基准调整网格以提高对温度场的计算精度。

2.1 问题概述

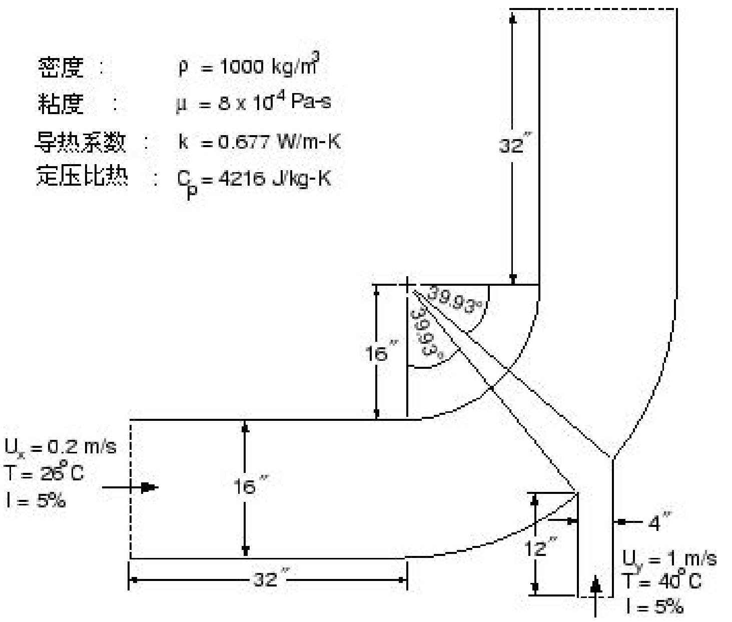


图 2-1 弯管流动图示

如图 2-1 所示，温度为 26℃的冷流体流过弯管，温度为 40℃的热流体从转弯处流入，

并与主流中的冷流体混合。管道的尺寸如图 2-1 所示，单位为英寸，而边界条件和流体材料性质则采用国际单位制。入口处的雷诺数为 2.03 x 10^5，因此必须使用湍流模型。

2.2 处理网格

网格处理包括网格的输入、检查、光顺、比例转换和显示等操作，下面分别进行介绍。

2.2.1 读入网格文件

首先启动 FLUENT 的 2D 版，然后读入网格文件：

File -> Read -> Case...

这个算例的网格文件可以在 FLUENT6.1 为用户提供的文档光盘中找到，路径是：

cdrom:\fluent6.1\help\tutfiles\elbow\elbow.msh

2.2.2 检查网格

执行下列菜单操作，进行网格检查：

Grid -> Check

此时控制台窗口中会显示与网格有关的信息，包括网格空间范围、体积信息、表面积信息、节点信息等等。网格中存在的任何错误都会出现在这个信息报告中，其中最需要检查的是网格单元的体积不能为负值，否则计算将无法继续下去。

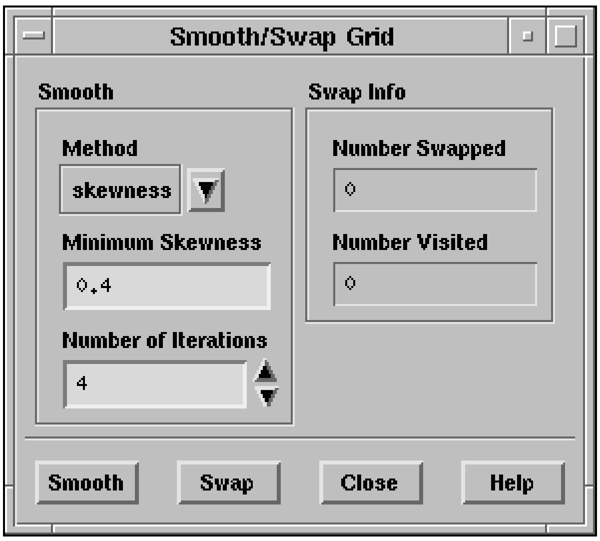


图 2-2 Smooth/Swap Grid（光顺/转换网格）面板

2.2.3 光顺并转换网格

执行下列菜单操作，打开 Smooth/Swap（网格光顺和转换）面板：

Grid -> Smooth/Swap...

光顺网格可以提高网格质量，提高计算精度。在启动光顺/转换网格面板后，点击 Smooth（光顺）按钮，然后再点击 Swap（转换）按钮，控制台上将报告被转换网格的数量。反复点击 Smooth（光顺）按钮和 Swap（转换）按钮，直到报告中被转换（Swap）的网格数量降低为零，则网格光顺处理过程结束。

2.2.4 按比例调整网格

执行下列菜单操作，打开 Scale Grid（按比例调整网格）面板：

Grid -> Scale...

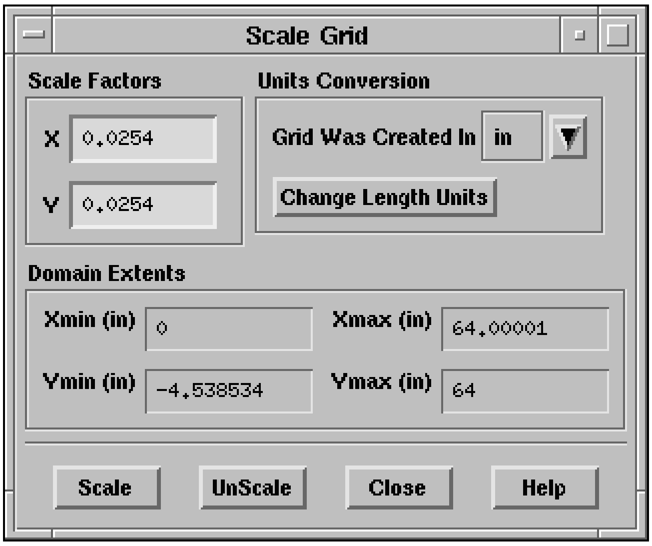


图 2-3 Scale Grid（按比例转换网格）面板

在 Units Conversion（单位转换）下面的 Grid Was Created In 列表中选择 in（英寸）作为网格创建数据单位。

点击 Scale（比例）按钮进行比例计算。注意在计算域范围（Domain Extents）中，网格的单位显示仍然是国际单位制中的 m（米）。

点击 Change Length Units（改变长度单位）按钮，可以将英寸变为长度单位，同时可以从面板下方计算域范围中看到 x、y 坐标的最大值改变为 64 英寸。

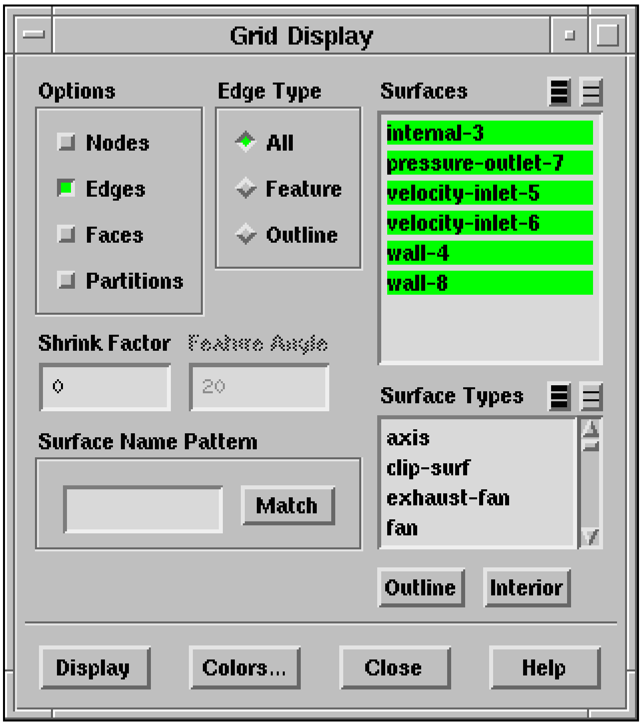


图 2-4 Grid Display（网格显示）面板

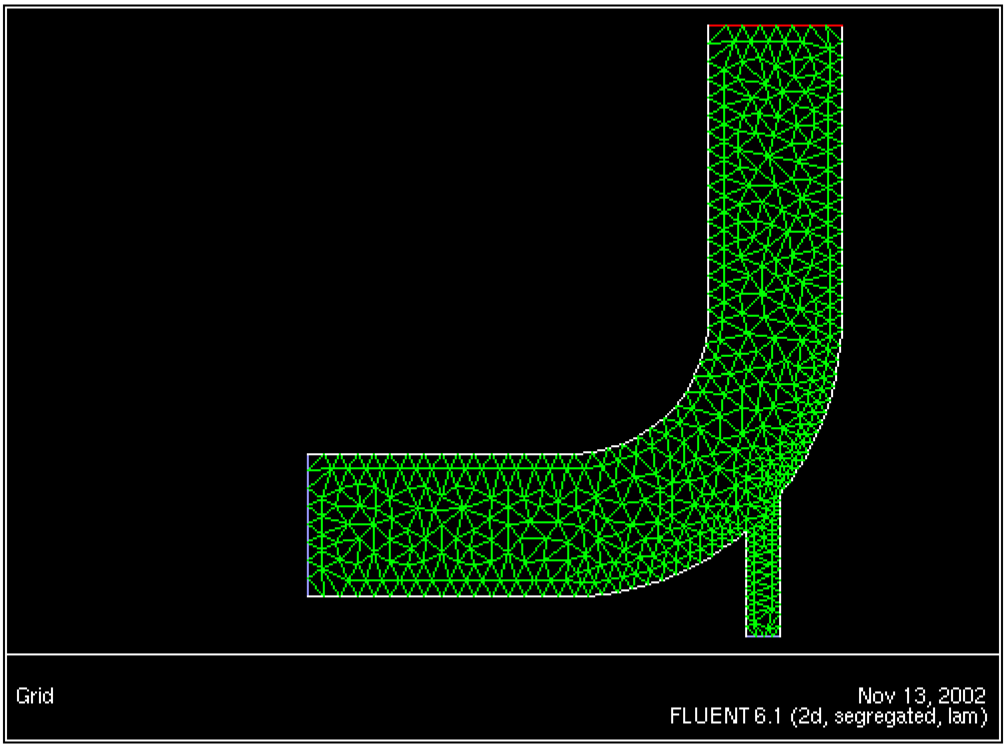


图 2-5 网格显示

2.2.5 显示网格

执行下列菜单操作，打开 Grid Display（网格显示）面板，如图 2-4 所示：

Display -> Grid...

选择所有表面，然后点击 Display（显示）按钮，FLUENT 随即打开一个新的显示窗口显示网格，如图 2-5 所示。

用右键点击图形显示窗口中的一个边界，则与这个边界相关的信息就会显示在控制台窗口中。在网格区域很多时，用这种方法可以比较方便地分辨各个区域。

2.3 计算模型

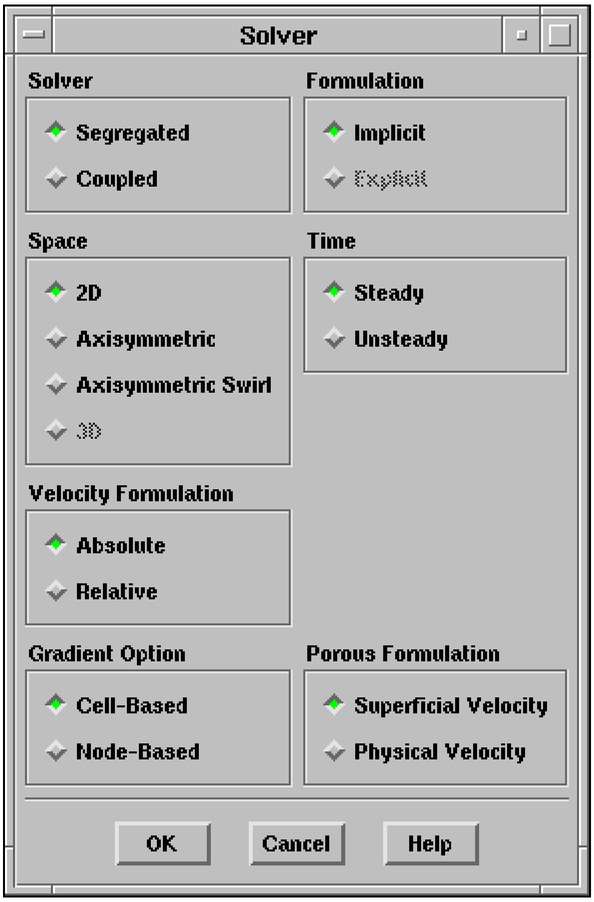


图 2-6 Solver（求解器）面板

2.3.1 设置求解器参数

执行下列菜单操作，打开 Solver（求解器）面板：
Define -> Models->Solver...
求解器的缺省设置如图 2-6 所示，本算例中可以保持这些缺省参数设置。

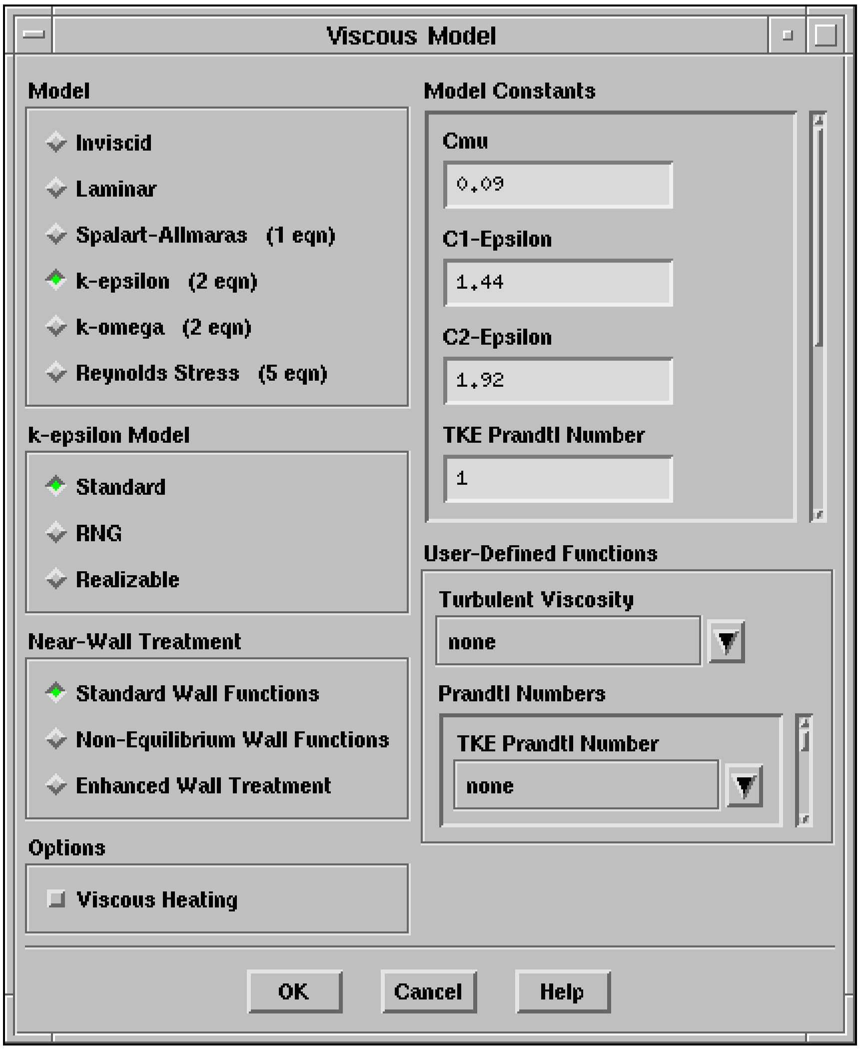


图 2-7 粘性模型面板

2.3.2 选择湍流模型

执行下列菜单操作，打开 Vicous（粘性）面板，如图 2-7 所示：

Define->Models->Viscous...

在 Model 列表中选择 k-epsilon (2 eqn)，即 $k-\varepsilon$ 两方程模型。点击 OK 按钮接受系统对 $k-\varepsilon$ 模型的缺省设置，包括标准模型设置、壁面函数设置和湍流模型常数的设置等等。

2.3.3 启用能量方程计算

执行下列菜单操作，打开 Energy（能量）面板，并激活能量计算，如图 2-8 所示：

Define->Models->Energy...

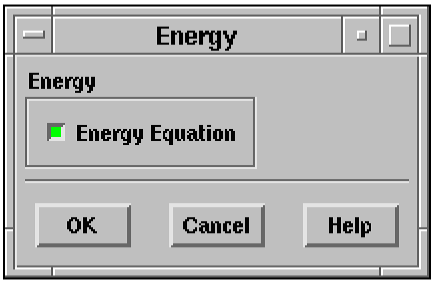


图 2-8 能量方程面板

2.4 定义材料性质

创建一种新的物质，命名为 water（水），并设置相关参数。

首先执行下列菜单操作，打开 Materials（材料）面板，如图 2-9 所示：

Define -> Materials...

在 Name（名称）中键入材料名称 water（水），然后设定水的各种物理性质，其中 Density（密度）为 1000kg/m^3， C_p（定压比热）为 4216J/kg-K，thermal conductivity（导热系数）为 0.677W/m-K，viscosity（粘度）为 8x10^{-4}kg/m-s。最后点击 Change/Create（修改或创建）按钮完成创建过程。在 FLUENT 提示是否用新的材料数据覆盖 air 的数据时，点击 NO 按钮予以否认。

实际上 FLUENT 的材料性质数据库中已经存在水的数据，因此也可以简单地将相关数据直接从 Database（数据库）中调入，如果个别数据与前面的数据有出入，可以直接在面板上进行修改。

设置完毕后，关闭 Materials（材料）面板。

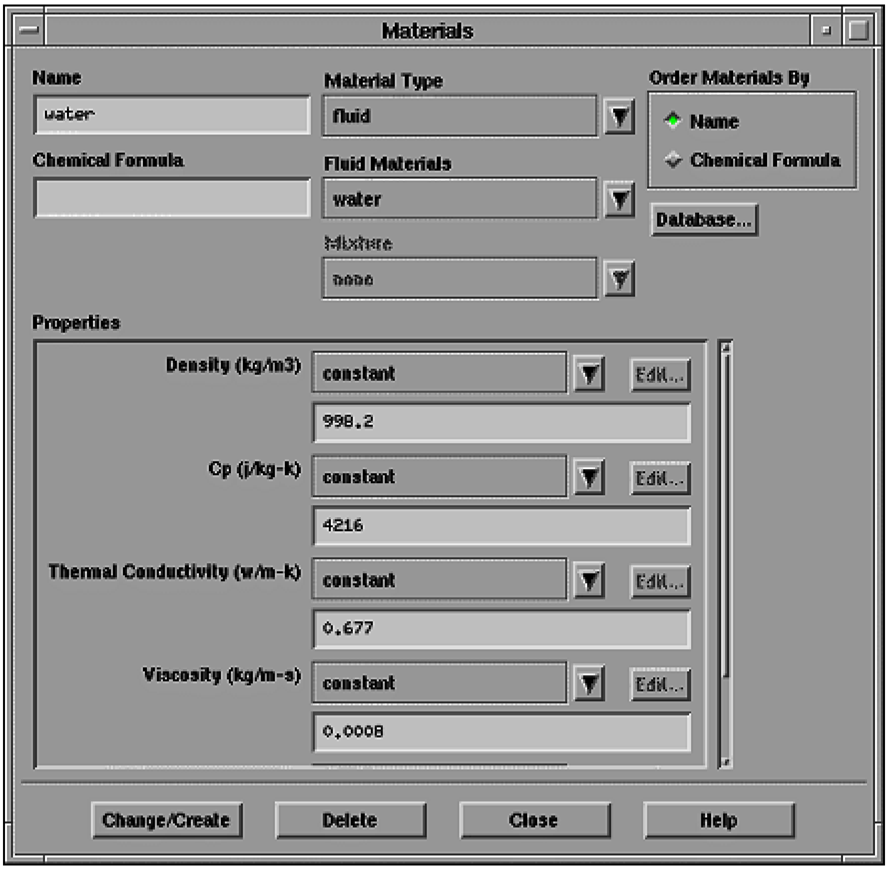


图 2-9 Materials（材料）面板

2.5 定义边界条件

执行下列菜单命令，打开边界条件面板，如图 2-10 所示：
Define -> Boundary Conditions...

2.5.1 设定流体条件

在 Zone（区域）列表中选择 fluid-9，Type（类型）列表中相应地反白显示为 fluid（流体）。

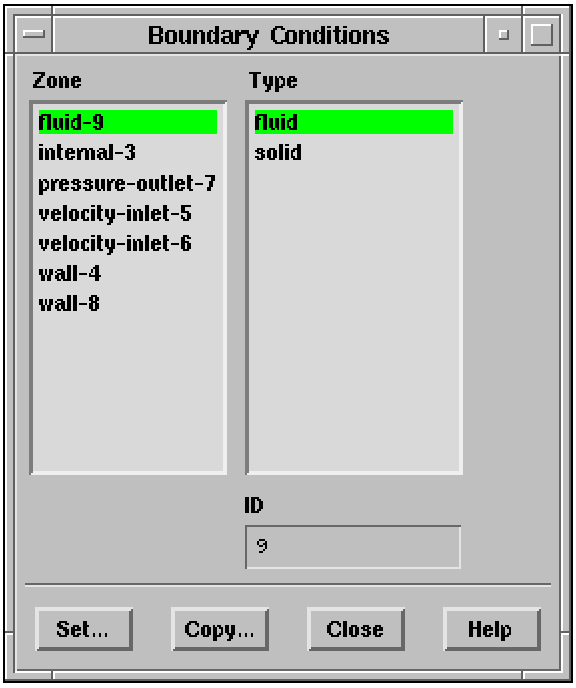


图 2-10 边界条件面板

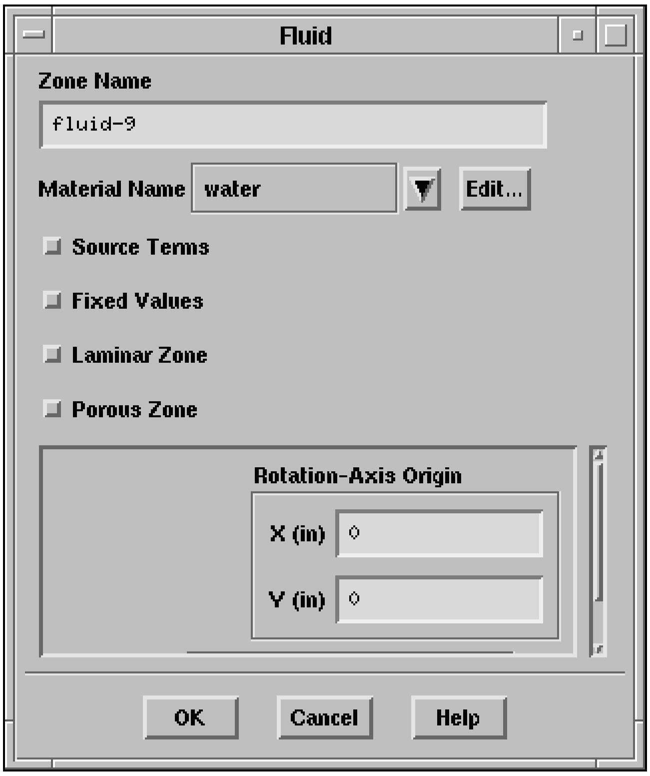


图 2-11 Fluid（流体）面板

点击 Set...（设置）按钮打开 Fluid（流体）面板，如图 2-11 所示。在 Material Name（材料名称）列表中选择 water（水），点击 OK 按钮结束设置过程。

2.5.2 设置速度入口边界条件。

（1）设置 velocity-inlet-5 上的边界条件。

在 Zone（区域）列表中选择 velocity-inlet-5，然后点击 Set...（设置）按钮，进入速度入口设置面板，如图 2-12 所示。

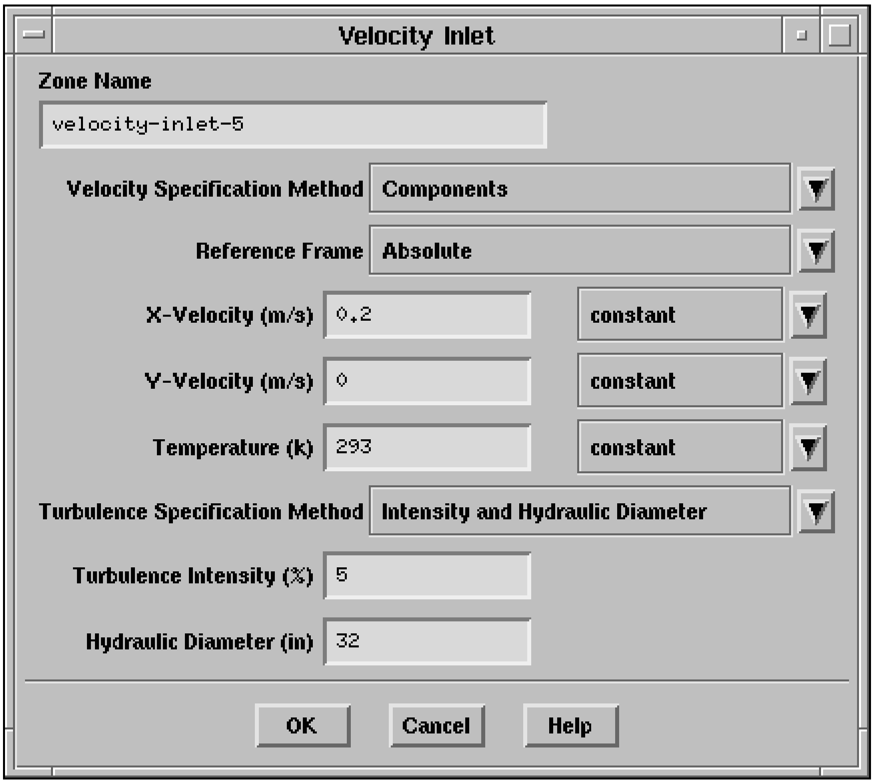


图 2-12 Velocity Inlet（速度入口）面板

在 Velocity Specification Method（速度定义方法）中，选择 Components（分量形式）定义速度，然后将 X-Velocity（X 向速度分量）设为 0.2m/s。将 Temperature（温度）设为 293K。在 Turbulence Specification Method（湍流定义方法）中，选择 Intensity and Hydraulic Diameter（强度和水力直径）定义湍流，然后将 Turbulence Intensity（湍流强度）设为 5%，将 Hydraulic Diameter（水力直径）设为 32in。点击 OK 按钮结束对 velocity-inlet-5 的设置。

（2）设置 velocity-inlet-6 上的边界条件，方法同上，相关参数如下：Y-Velocity（Y 向速度分量）为 1.0m/s，temperature（温度）为 313K，Turbulence Intensity（湍流强度）为 5%，Hydraulic Diameter（水力直径）为 8in（英寸）。

2.5.3 设置压强出口边界条件

pressure-outlet-7 上的压强出口边界条件的相关参数为：表压 0 pascal，回流总温为 300 K，Backflow Direction Specification Method（回流方向定义方法）为 Normal to Boundary（垂直于边界），Turbulence Specification Method（湍流定义方法）为 Intensity and Hydraulic Diameter（湍流强度与水力直径），Backflow Turbulence Intensity（回流湍流强度）为 5％，Backflow Hydraulic Diameter（回流水力直径）为 32in。

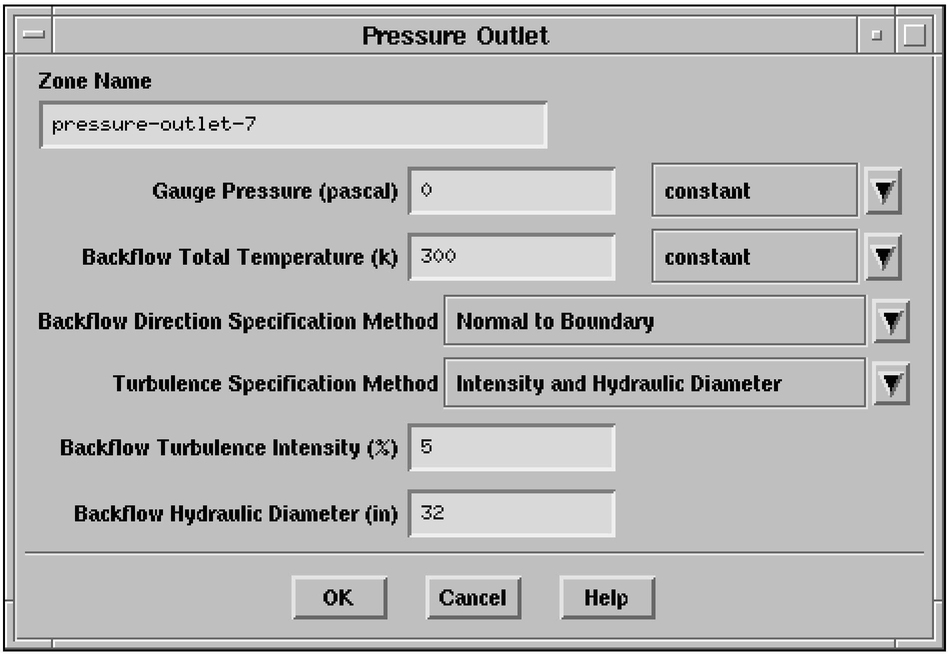


图 2-13 Pressure Outlet（压强出口）面板

2.5.4 壁面条件设置

壁面 wall-4 的边界条件保持缺省设置，其中 Heat Flux（热流通量）的值为 0。壁面 wall-8 也保持缺省设置。

2.6 求解过程

参数设置完毕并检查无误后，既可以开始流场计算。流场计算的流程如下：

2.6.1 初始化

执行下列菜单操作打开 Solution Intialization（求解初始化）面板，如图 2-14 所示：
Solve->Initialize->Initialize...

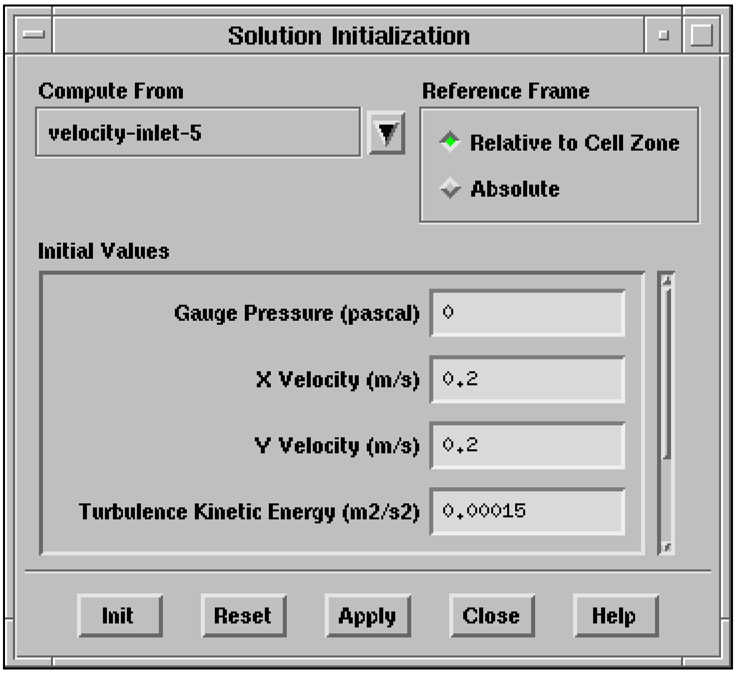


图 2-14 Solution Initialization（初始化解）面板

在 Compute From（计算起始于）列表中选择 velocity-inlet-5，然后在 Y Velocity（Y 向速度分量）中加入 0.2m/sec，点击 Init（初始化）按钮开始初始化，最后点击 Close（关闭）按钮关闭初始化面板。

2.6.2 设置残差监视器

执行下列菜单操作，打开 Residual Monitors（残差监视器）面板：
Solve->Monitors->Residual...
在 Options（选项）下选择 Plot（绘图），点击 OK 按钮关闭面板。

2.6.3 保存算例文件

执行下列菜单操作保存算例文件：

File->Write->Case...

不用改动 Write Binary Files（写二进制文件）的缺省设置，这样算例文件可以用二进制格式记录在硬盘上。

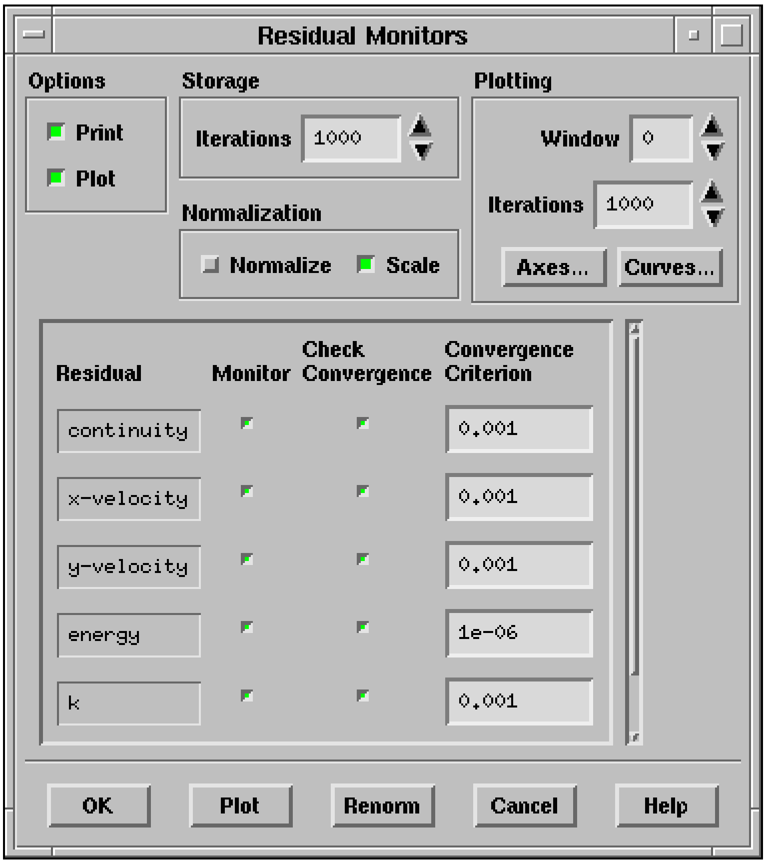


图 2-15 Residual Monitors（残差监视器）面板

2.6.4 打开迭代面板

执行下列菜单操作，打开 Iterate（迭代）面板：

Solve -> Iterate...

将 Number of Iterations（迭代次数）设为 100，点击 Iterate（迭代）按钮开始计算。在开始计算时残差监视器将同时被打开。

2.6.5 监视收敛过程

图 2-16 是计算进行到第 60 步时残差曲线的走势。因为没有普适的收敛判断标准，所

以在观察残差曲线时，不要仅仅监视残差曲线下降的数量级，最好同时能够监视相关流场变量的变化情况。

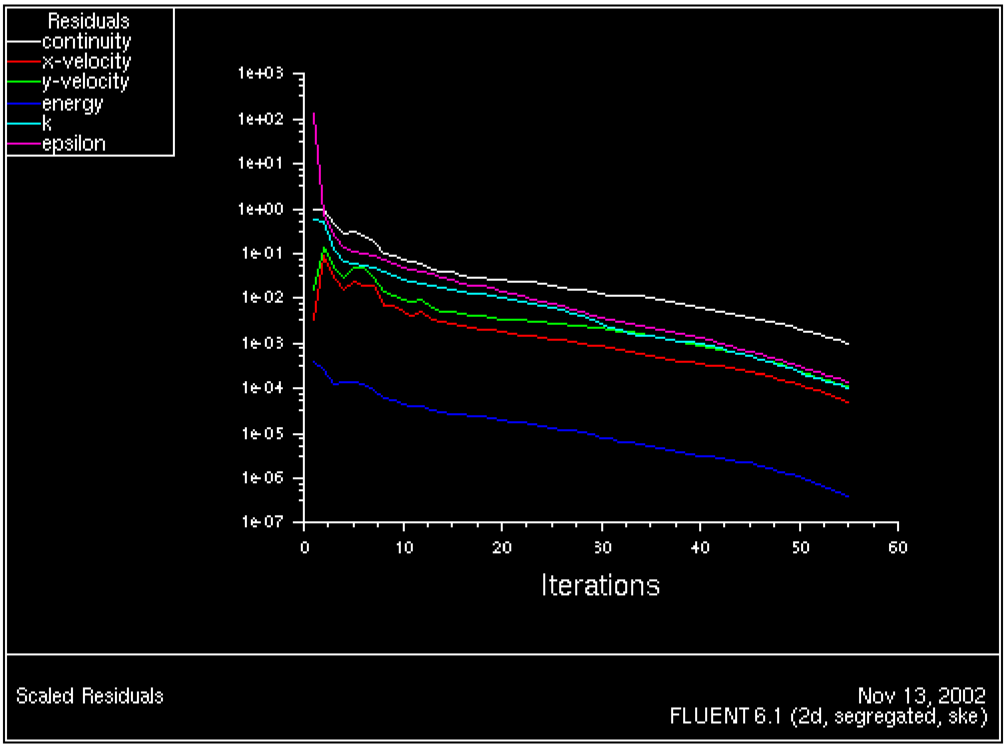


图 2-16 残差监视器

简单地说，可以用三种方法判断计算是否已经收敛：

（1）观察残差曲线。

可以在残差监视器面板中设置 Convergence Criterion（收敛判据），比如设为 10^{-3}，则残差下降到小于 10^{-3} 时，系统既认为计算已经收敛并同时终止计算。

（2）流场变量不再变化。

有时候不论怎样计算，残差都不能降到收敛判据以下。此时可以用具有代表性的流场变量来判断计算是否已经收敛——如果流场变量在经过很多次迭代后不再发生变化，就可以认为计算已经收敛。

（3）总体质量、动量、能量达到平衡。

在 Flux Reports（通量报告）面板中检查质量、动量、能量和其他变量的总体平衡情况。通过计算域的净通量应该小于 0.1%。Flux Reports（通量报告）面板如图 2-17 所示，其启动方法为：

Report -> Fluxes

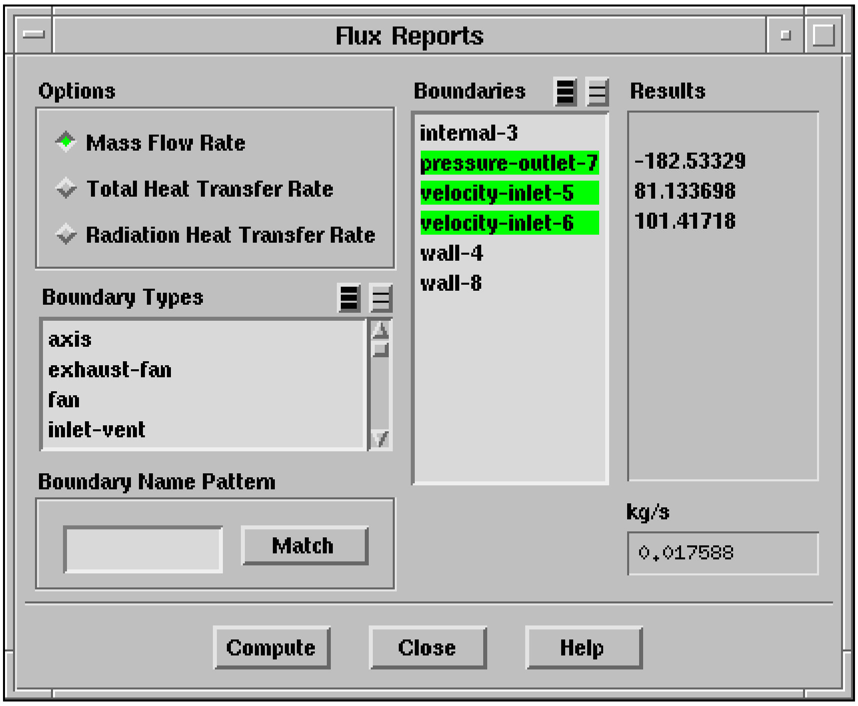


图 2-17 Flux Reports（通量报告）面板

2.6.6 保存数据文件

执行下列菜单操作保存数据文件：

File -> Write -> Data...

2.7 显示计算结果

2.7.1 等值线显示

执行下列菜单操作，启动 Contours（等值线）面板，如图 2-18 所示：

Display -> Contours...

首先显示速度值的填充等值线。在 Contours of 下的列表中选择 Velocity...（速度）和 Velocity Magnitude（速度值），然后在 Options（选项）下面选择 Filled（填充），最后点击 Display（显示）按钮显示速度的等值线。在等值线图中用右键点击一点，控制台窗口中将显示该点的速度值。

用同样的方法可以显示温度、压强等变量的等值线图。

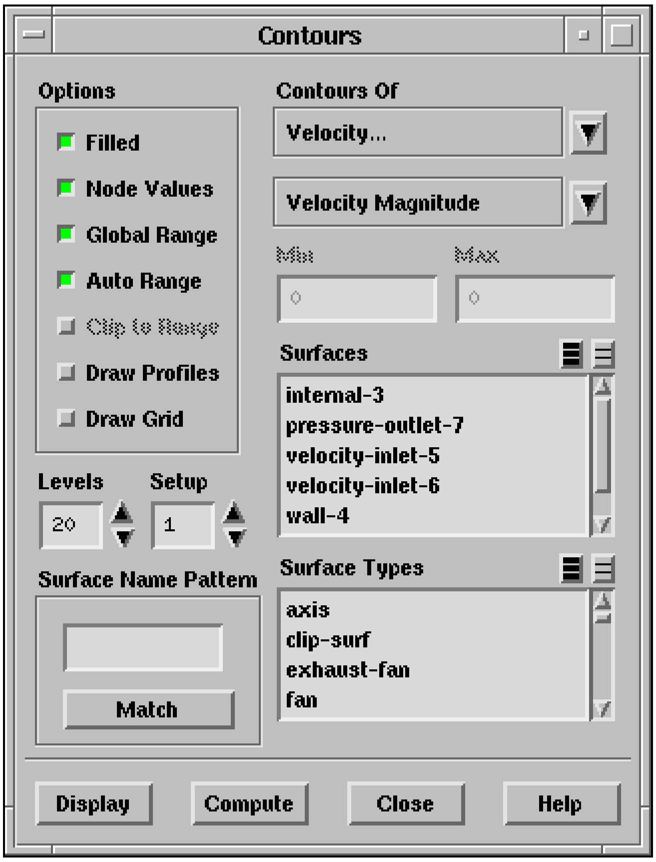


图 2-18 Contours（等值线）面板

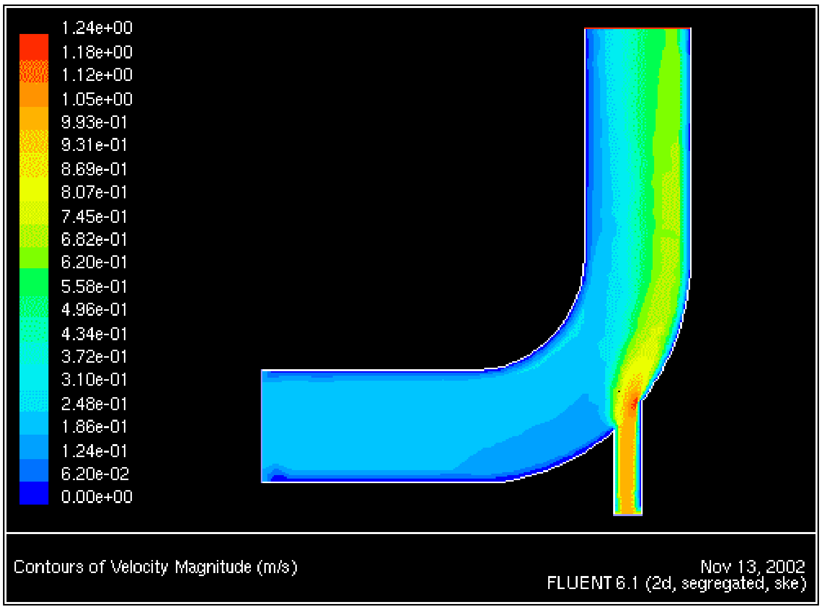


图 2-19 速度值的等值线图

2.7.2 显示矢量图

除了显示等值线图，还可以用矢量图观察流场中的速度矢量。

首先执行下列菜单命令，启动 Vectors（矢量）面板：

Display -> Vectors...

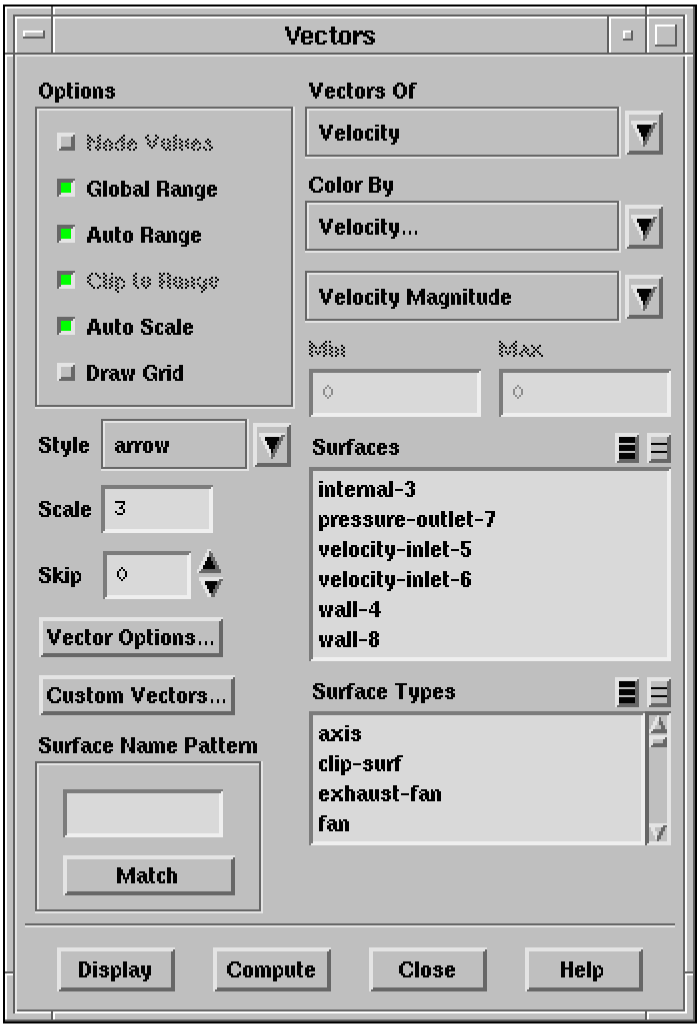


图 2-20 Vectors（矢量）面板

在 Vectors of 下选择 Velocity（速度），然后点击 Display（显示）按钮，就可以在不改变缺省设置的情况下显示速度矢量图。如果对显示效果不满意，可以在 Vectors（矢量）面板中修改显示设置，包括改变 Scale（比例）和 Style（风格）。

如果计算的是三维问题，则需要先在 Surfaces（表面）下选择一个显示面，才能进行显示，显示的结果是速度矢量在这个面上的投影。

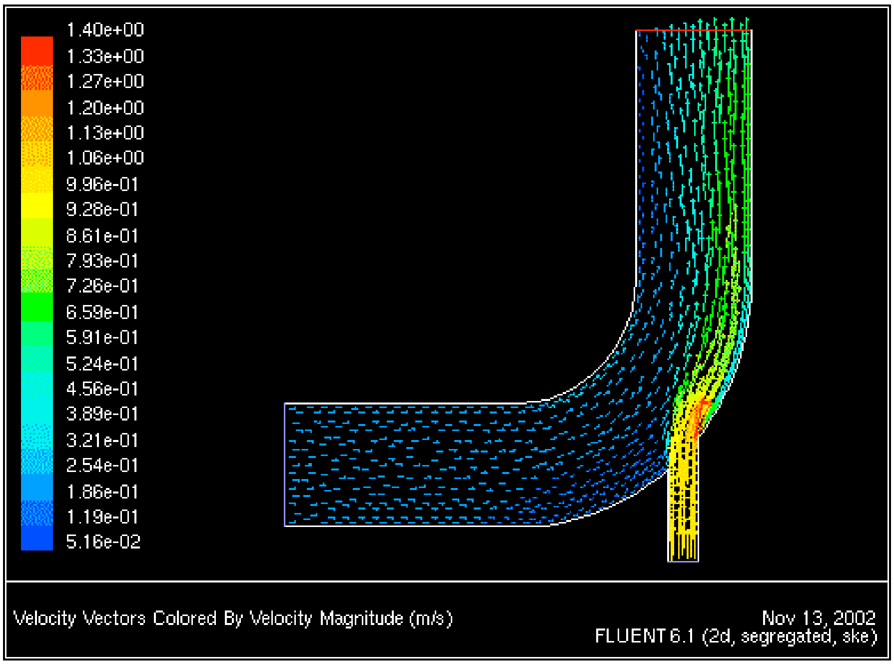


图 2-21 速度矢量图

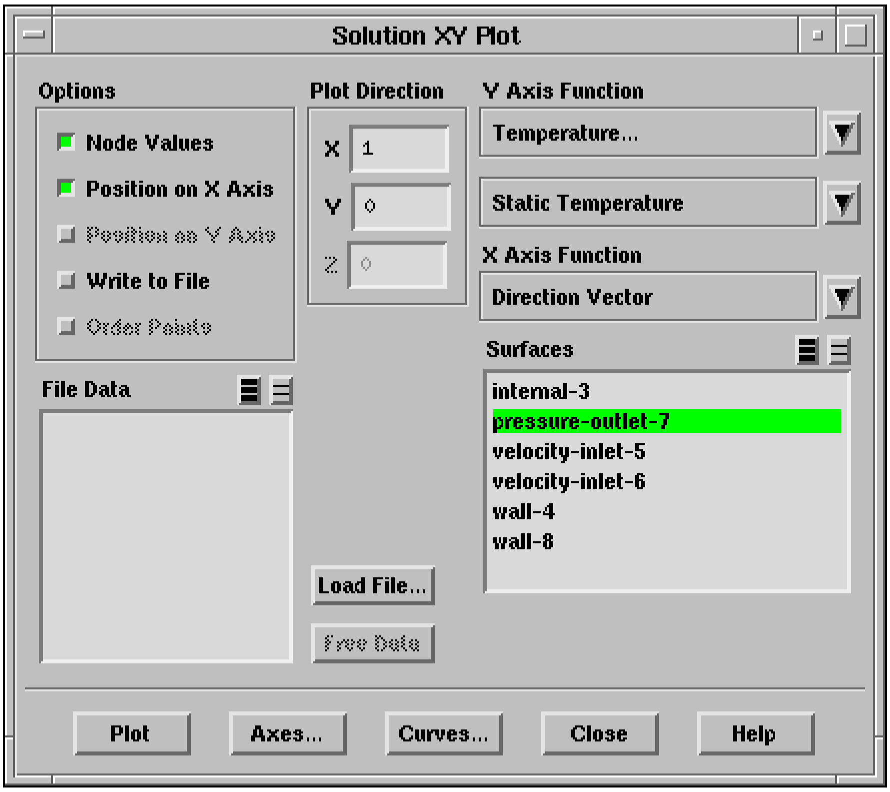


图 2-22 解变量 XY 绘图面板

2.7.3 XY 曲线显示

出口处的温度变化可以用 XY 曲线形式予以显示。首先启动 Solution XY Plot（解变量 XY 绘图）面板：

Plot -> XY Plot...

在 Y Axis Function（Y 轴函数）下面的列表中选择 Temperature...（温度）和 Static Temperature（静温）作为 Y 轴变量，在 Surfaces（表面）列表中选择 pressure-outlet-7，点击 Plot（绘制）按钮，则静温在出口处的分布曲线就显示在图形窗口中。用同样方法还可以显示压强等变量在出口、入口等处的分布曲线。

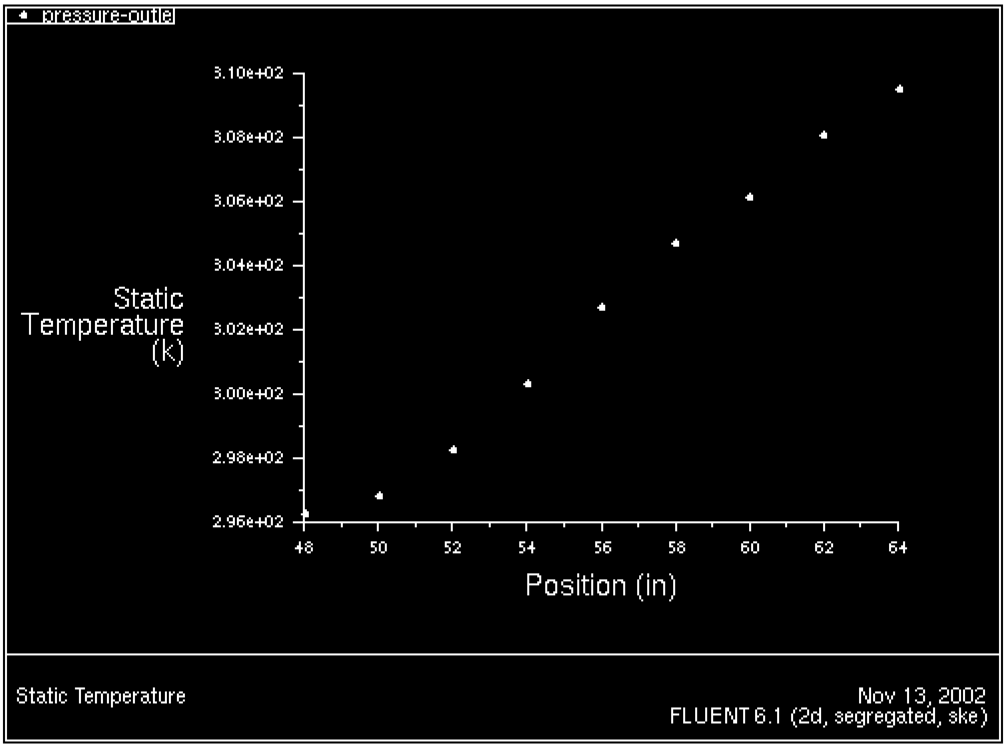


图 2-23 温度在出口处的分布曲线

2.7.4 定义场函数

定义动压头的场函数。首先启动 Custom Field Function Calculator（场函数计算器）面板：

Define -> Custom Field Functions...

将新函数的名称加入 New Function Name（新函数名称）栏，然后根据动压头 $\rho \cdot |v|^2 / 2$

的定义公式进行公式编辑。动压头的输入内容如图 2-24 所示。

在公式中需要输入流场变量（比如密度、速度）时，就在 Field Functions（场函数）列表下进行选择；在使用数字和运算操作符的时候，则直接从面板左部的键盘中点取。输入完毕后，点击 Define（定义）按钮完成函数定义。

场函数的显示与其他场变量的显示方法是完全相同的。动压头的显示与速度值的显示方法相同：

（1）启动等值线面板。

（2）然后从 Contours of 列表中选择 Custom Field Function...（修改场函数）和 dynam-head（动压头）。

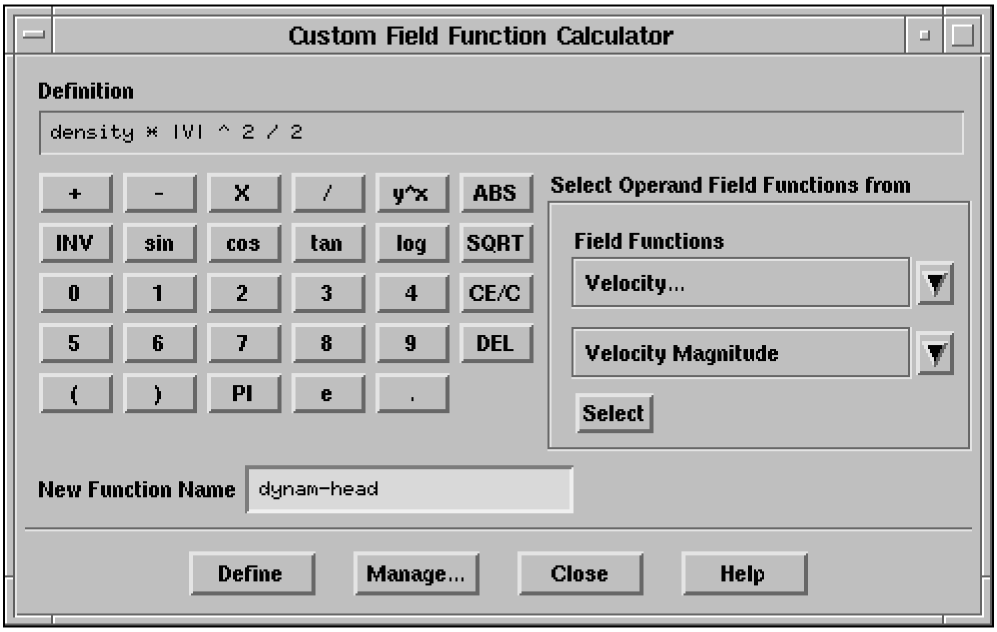


图 2-24 场函数计算器面板

（3）设置显示选项，点击 Display（显示）按钮。

2.8 启用二阶精度离散格式

前面的计算是用一阶精度格式完成的。一阶精度格式的缺点是耗散性很大，计算稳定性好，但是对流场中梯度比较大区域内的解有比较严重的“抹平”现象，因此为了获得精度更高的结果，可以采用二阶精度格式。因为二阶精度格式的稳定性不如一阶精度，所以在采用二阶精度格式的时候要适当减小亚松弛因子。二阶精度格式的具体设置和求解过程如下：

（1）启动 Solution Controls（求解控制）面板：

Solve -> Controls -> Solution...

在 Discretization（离散）下面的 Energy（能量）旁边的列表中选择 Second Order（二阶精度）。在 Under-Relaxation Factors（亚松弛因子）下面的 Energy（能量）旁选择 0.8，如图 2-25 所示。

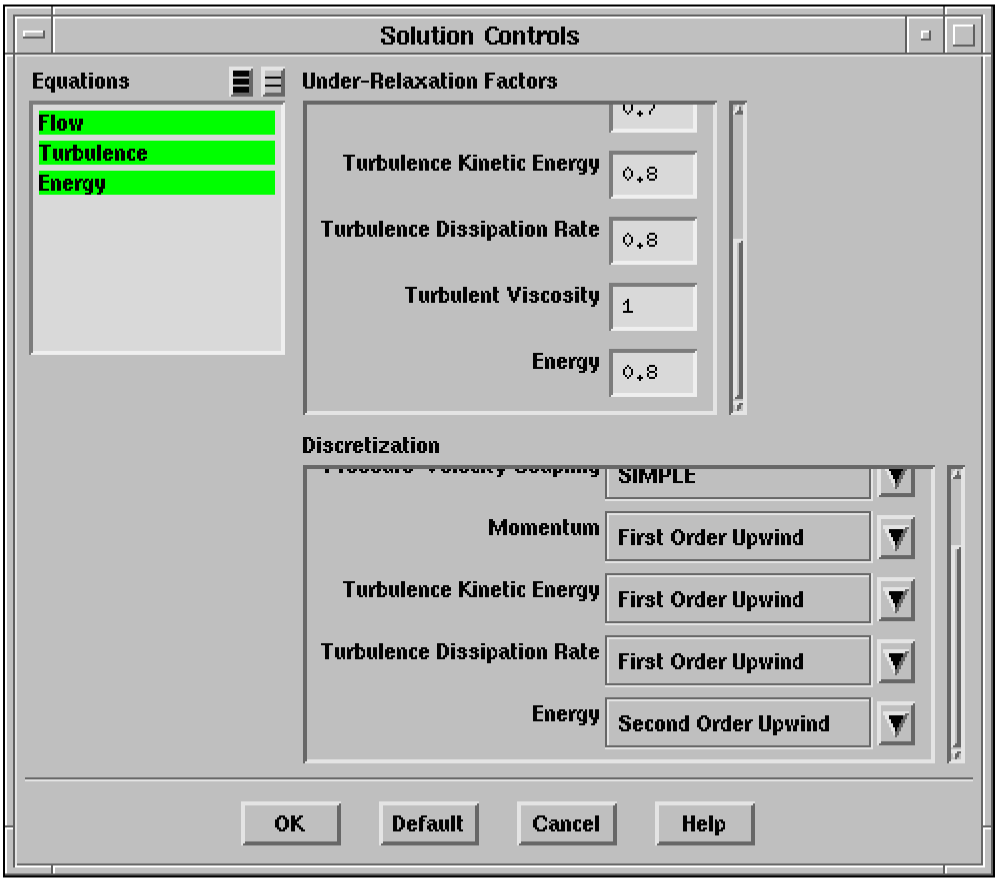


图 2-25 修改求解控制参数

（2）在 Iterate（迭代）面板中，将 Number of Iterations（迭代步数）的值修改为 100，点击 Iterate（迭代）按钮，继续进行 100 次计算，则残差曲线如图 2-26 所示。

（3）执行下列菜单操作，保存算例文件和数据文件：

File -> Write -> Case & Data...

（4）再次显示温度等值线，如图 2-27 所示。

图 2-28 是一阶精度计算的温度场等值线。对比一阶精度计算和二阶精度计算的等值线图，可以发现在采用一阶精度进行计算的流场中，冷热流的混合区大于二阶精度中的混合区，也就是说一阶精度计算对冷热流的混合过程模拟的误差大于二阶精度格式，计算流场被“抹平”了。

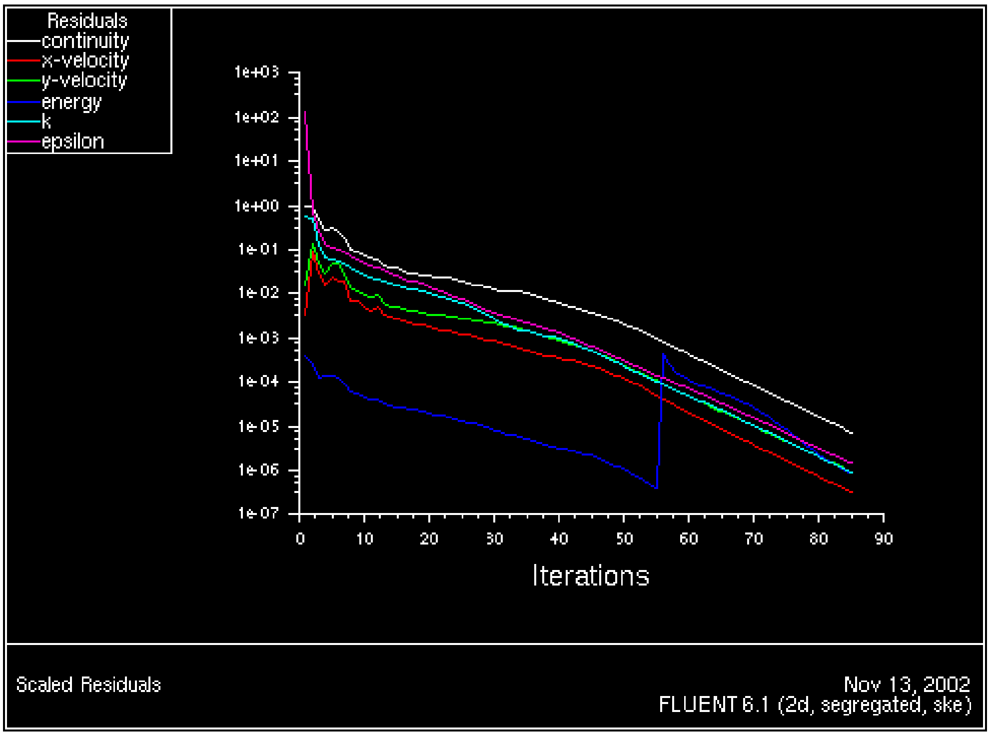


图 2-26 改为二阶精度后的残差曲线

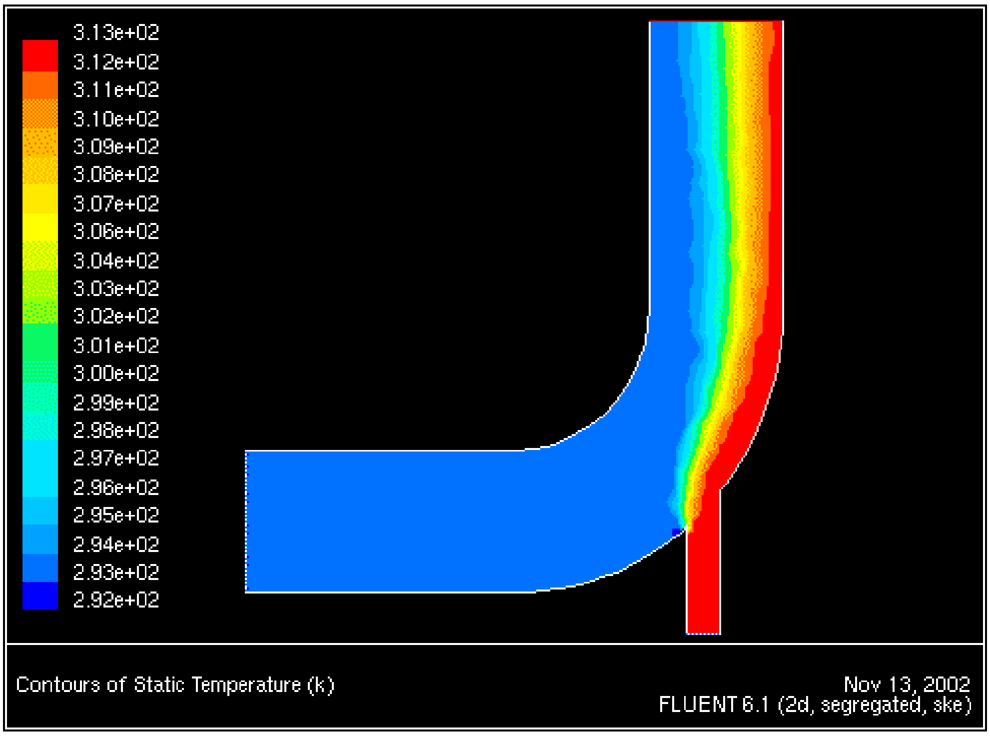


图 2-27 二阶精度格式得到的温度场

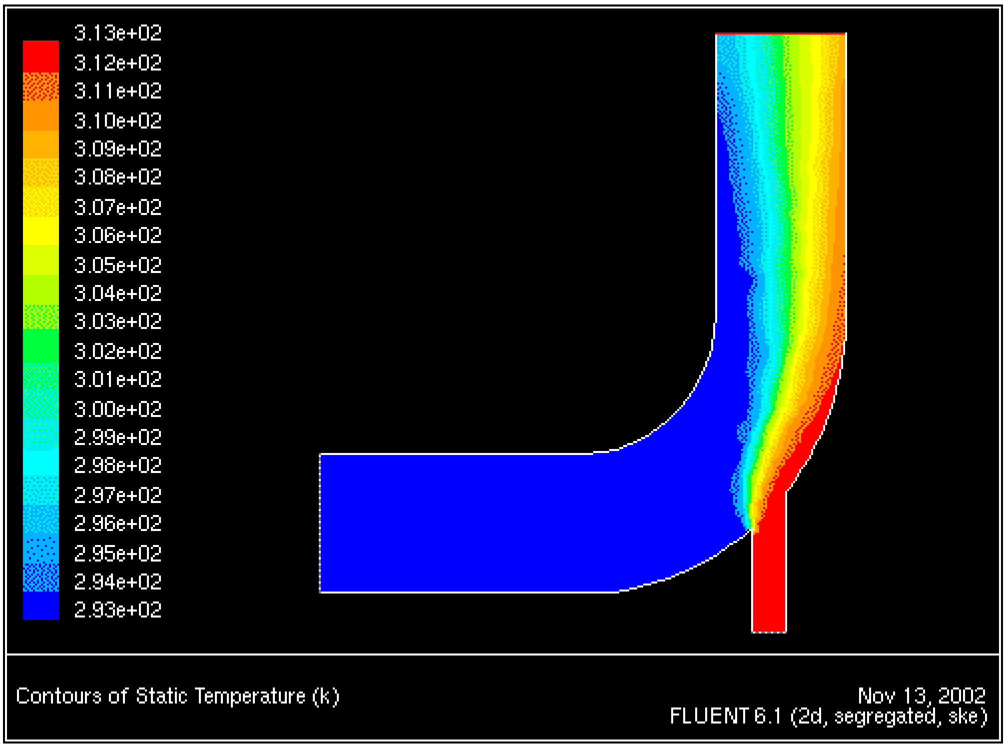


图 2-28 一阶精度计算的温度等值线

2.9 调整网格

如果适当调整网格可以进一步提高计算精度。本小节根据前面计算的温度梯度对网格进行适应性调整，调整前先确定需要调整的梯度范围。在网格被重新划分后，就可以继续计算过程了。

2.9.1 按网格显示温度场

在等值线面板中，将变量名设为 Temperature...（温度）和 Static Temperature（静温），并取消 Options（选项）下面的 Node Values（节点值），这样既可得到按网格显示的温度场，如图 2-29 所示。

图 2-27 中看到过的温度等值线是在整个流场中进行计算得到的，而图 2-29 中的等值线图则是将网格中心点的值作为网格单元的值，逐个网格绘制得到的。

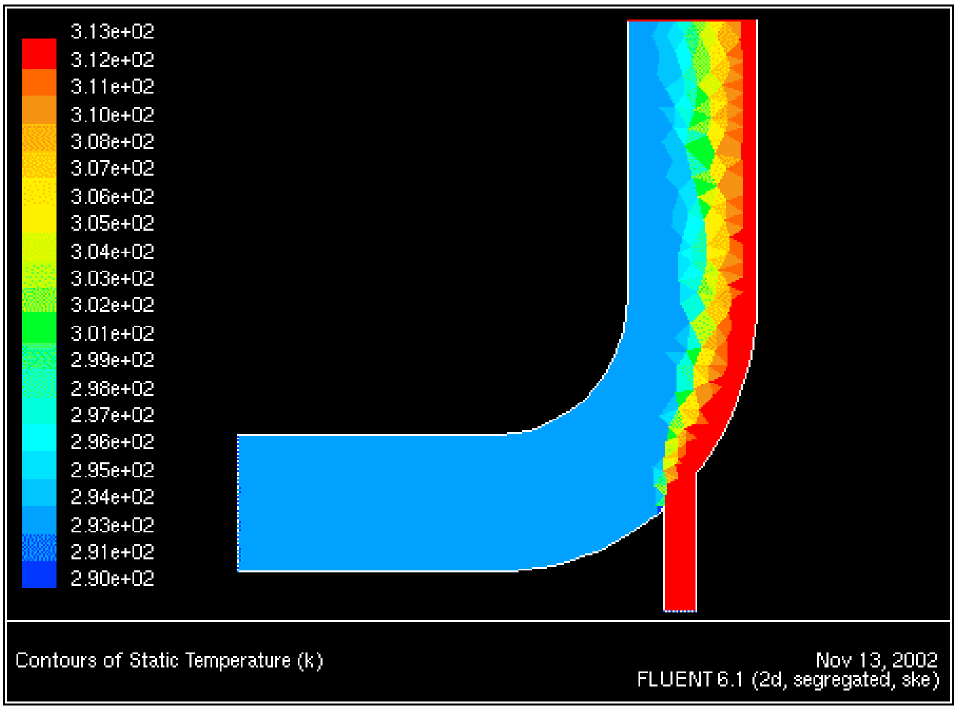


图 2-29 按网格显示的温度等值线

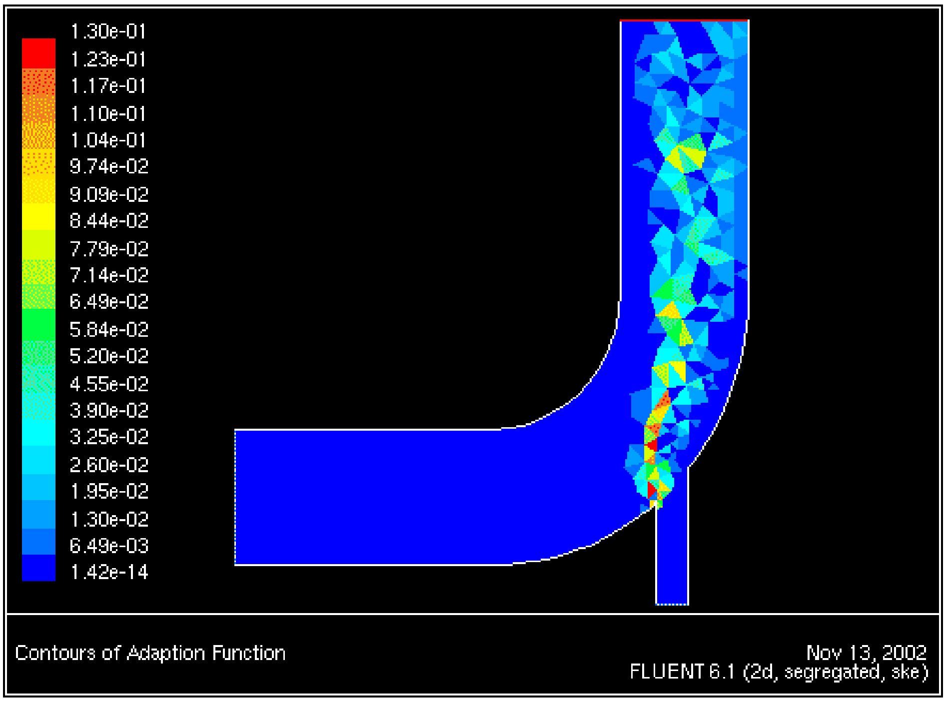


图 2-30 按网格显示的温度梯度场

2.9.2 按网格显示温度梯度场

在等值线面板中的 Contours Of 下面选择 Adaption...（适应）和 Adaption Function（适应函数），点击 Display（显示）按钮显示按网格显示的温度梯度场，如图 2-30 所示。

需要注意的是，Adaption Function（适应性函数）的值被系统缺省设定为最近显示过的函数的梯度值，在这里就是温度的梯度值。

2.9.3 显示特定范围的梯度值

显示某个范围内的梯度值，以便标示出需要调整的网格：

在 Options（选项）前面取消 Auto Range（自动范围）的选择，在 Min（最小值）中填入 0.02，即取消系统对显示范围的缺省设置，然后将显示的最小值设为 0.02。点击 Display（显示）按钮进行显示，结果如图 2-30 所示。

用这种方式可以显示梯度较高，因而需要进行调整的网格区域。

2.9.4 调整高梯度区域的网格

首先启动 Gradient Adaption（梯度适应）面板：

Adapt -> Gradient...

在 Gradient Of 列表中选择 Temperature...（温度）和 Static Temperature（静温），取消 Options（选项）下面对 Coarsen（粗糙）的选择，点击 Compute（计算），FLUENT 随即更新 Min（最小值）和 Max（最大值）。

在 Refine Threshold（细化阈值）中输入 0.02。点击 Mark（标记）按钮，FLUENT 将在控制台窗口中报告将被调整的网格数目。

点击左面的 Manage...（管理）按钮，打开网格适应管理（Manage Adaption Registers）面板，显示将被调整的网格信息。点击 Display（显示）按钮显示将被调整的网格，如图 2-34 所示。点击 Adapt（适应）按钮，并在弹出的提示信息窗口中，点击 Yes 按钮完成调整。

需要说明的是，网格适应性调整有两种办法，一种是在 Manage Adaption Registers（管理适应注册记录）面板中点击 Adapt（适应）按钮进行调整。也可以在 Gradient Adaption（梯度适应）面板中进行调整。如果用 Gradient Adaption（适应）面板进行网格调整，FLUENT 将重新创建一个 adaption register（适应注册记录），因此用 Manage Adaption Registers（管理适应注册记录）面板进行适应性调整可以节省很多时间。

网格适应性调整结束后，关闭 Manage Adaption Registers（管理适应注册记录）面板和 Gradient Adaption（梯度适应）面板。

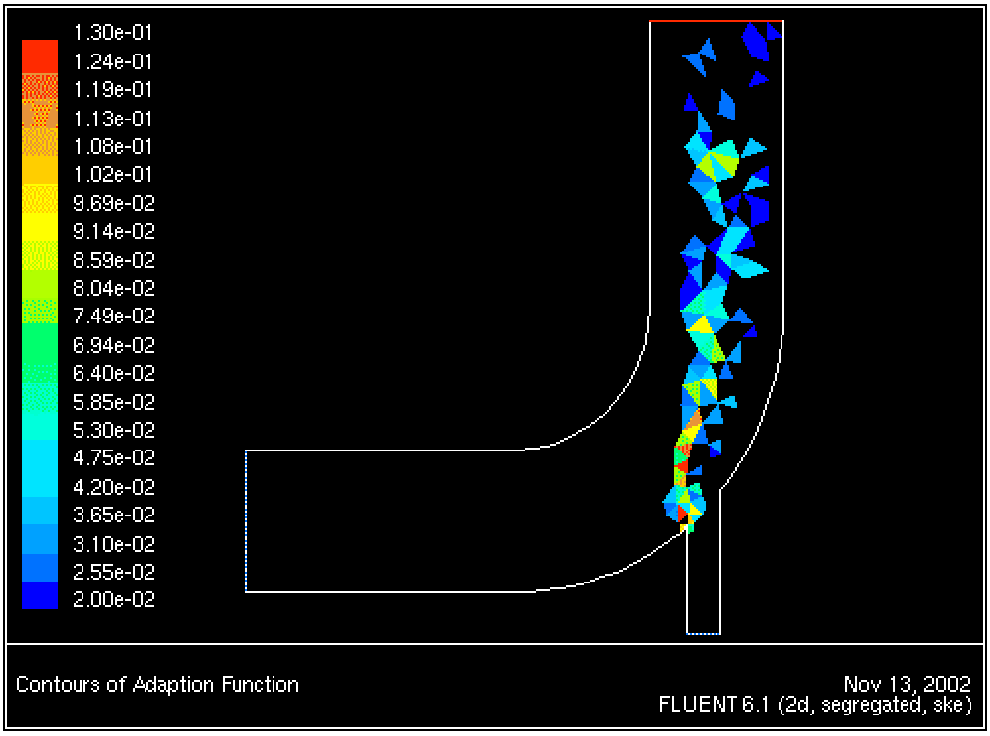


图 2-31 指定范围内的温度梯度场

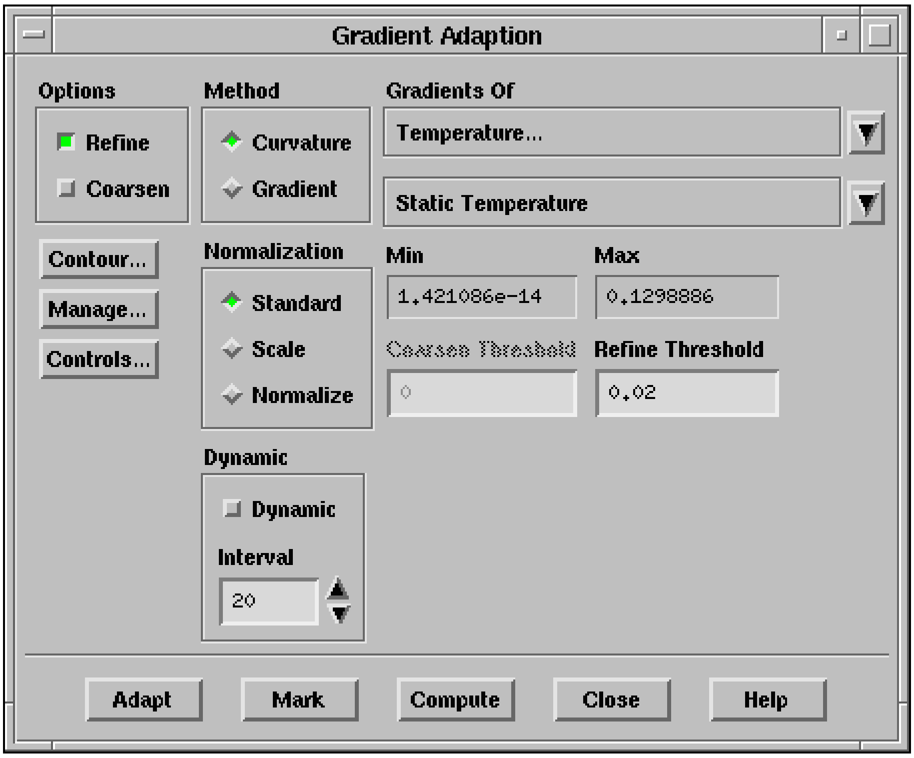


图 2-32 梯度调整面板

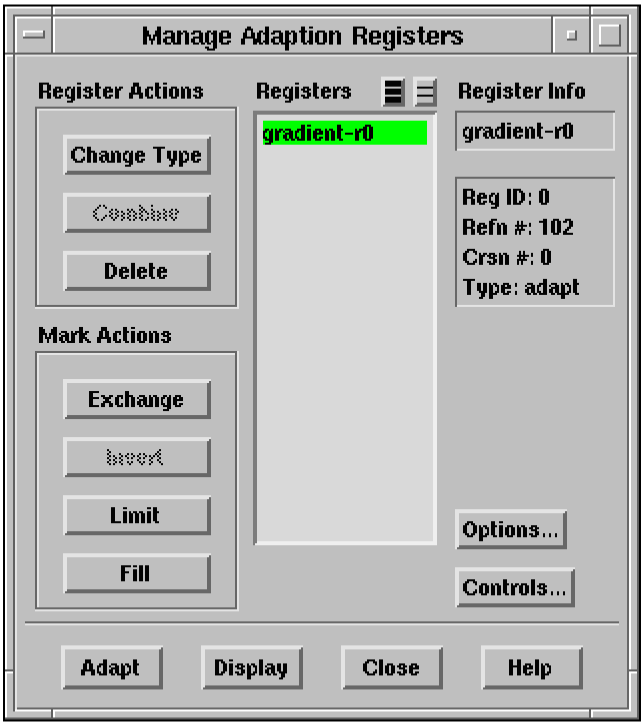


图 2-33 网格调整管理面板

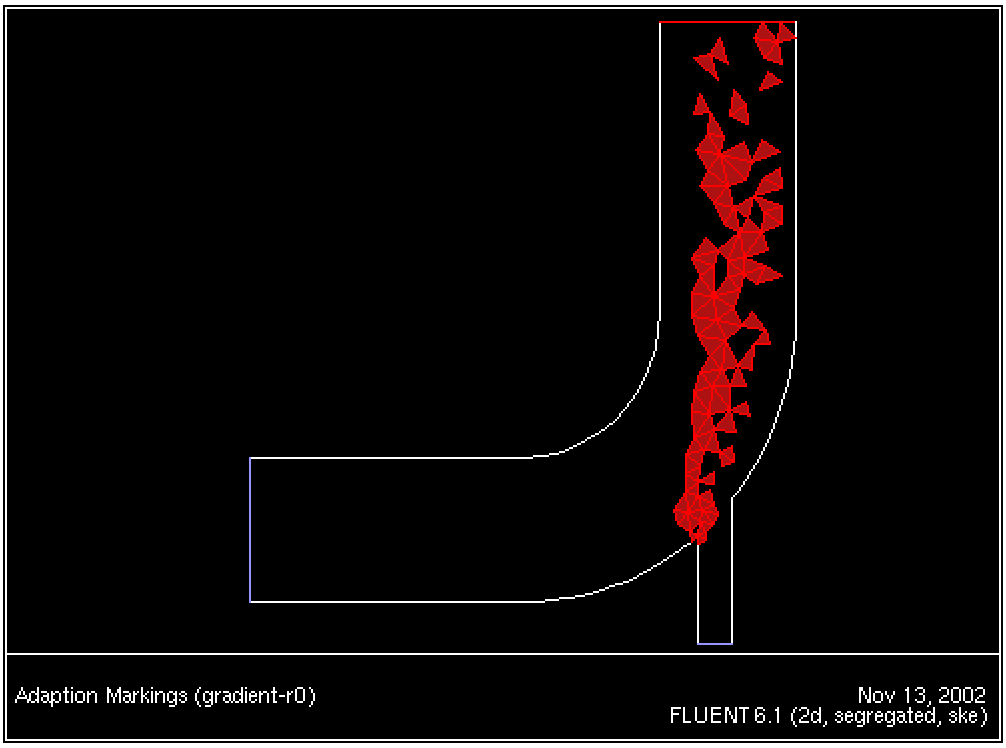


图 2-34 将被调整的网格

2.9.5 显示调整后的网格

执行下列菜单操作显示调整后的网格，如图 2-35 所示：
Display->Grid...

2.9.6 用调整后的网格进行计算

在前面的计算基础上，再进行 100 次迭代。在大约 40 步后，计算既收敛，如图 2-36 所示。计算结束后保存算例文件和数据文件。

2.9.7 检验计算结果

执行下列菜单操作打开 Contours...（等值线）面板，并显示温度等值线：
Display->Contours...
结果如图 2-36 所示，通过与前面结果的对比可以发现改进后的效果。

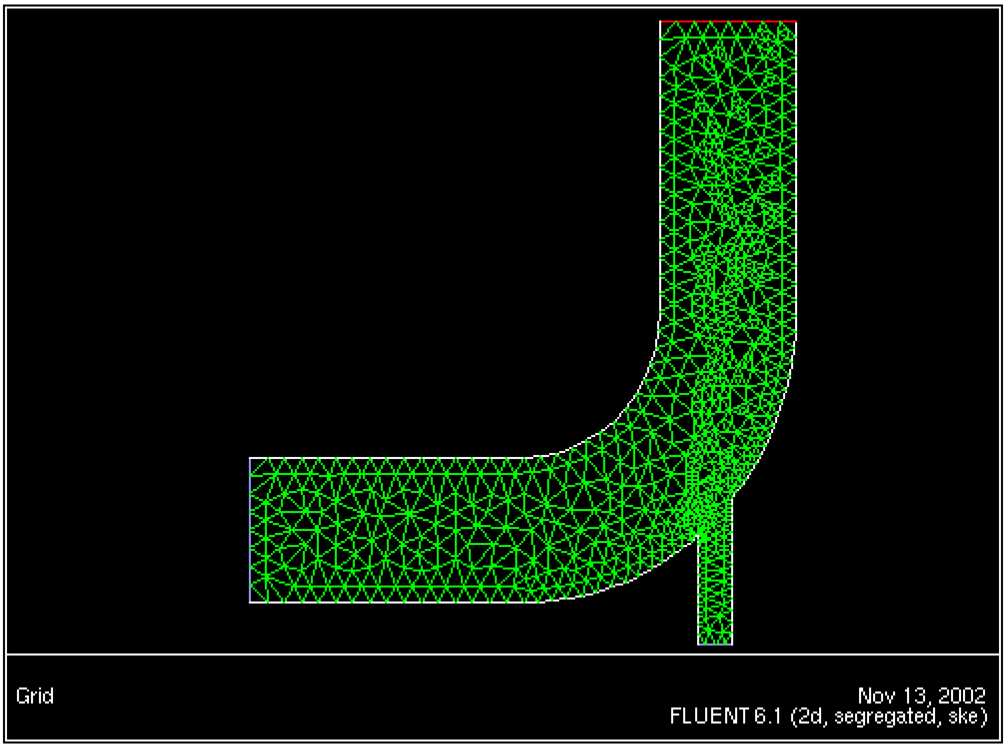


图 2-35 调整后的网格

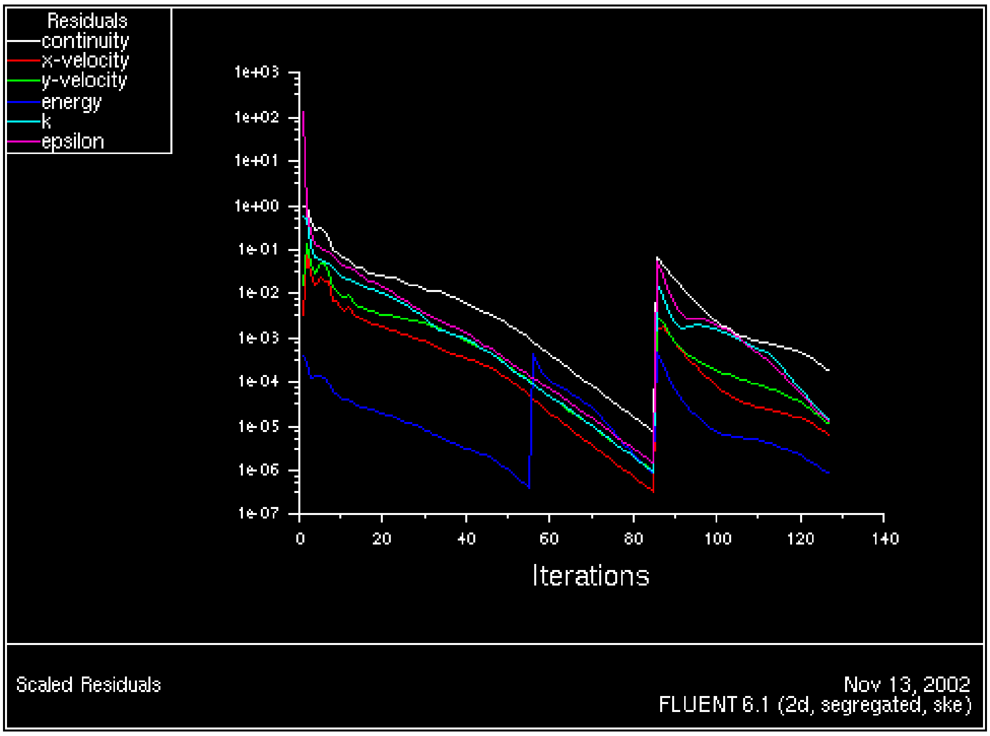


图 2-36 网格调整后的收敛曲线

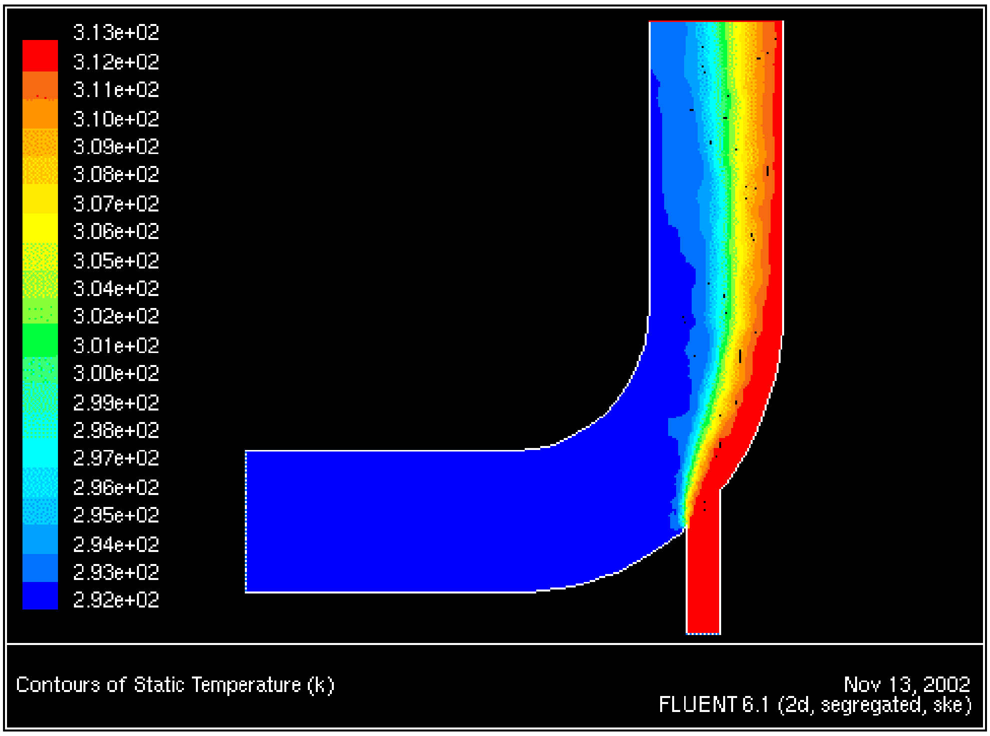


图 2-37 网格调整后的计算结果（温度场）

2.10 总结

通过对一阶精度的计算结果和采用适应性网格、并用二阶精度计算的结果进行对比，可以发现，后者的耗散性已经大大减小，计算精度得到提高。在 FLUENT 中，一阶精度格式是缺省设置的计算格式，在实际计算过程中可以用它获得初始流场，然后再提高计算格式精度，最后采用适应性网格技术。采用这样的计算策略，既可以保证计算的稳定性，又可以获得精度较高的流场计算结果，因此在复杂流场的计算中是经常使用的办法。

在本章的算例中，流场计算与温度场计算是分别进行的。在可能的情况下，将能量方程与其他控制方程分别求解，而不是耦合求解可以提高计算效率。在能量方程面板中，取消对能量方程的选择就可以将能量方程从流场中分离出来。

第二篇 预处理技术

第三章 GAMBIT 网格划分基础

GAMBIT 软件是 Fluent 公司提供的前处理器软件，它包含功能较强的几何建模能力和强大的网格划分工具，可以划分出包含边界层等 CFD 特殊要求的高质量的网格。GAMBIT 可以生成 FLUENT6、FLUENT5.5、FIDAP、POLYFLOW 等求解器所需要的网格。使用 Gambit 软件，将可大大缩短用户在 CFD 应用过程中建立几何模型和流场以及划分网格所需要的时间。用户可以直接使用 Gambit 软件建立复杂的实体模型，也可以从主流的 CAD/CAE 系统中直接读入数据。Gambit 软件高度自动化，可生成包括结构和非结构化的网格，也可以生成多种类型组成的混合网格。如果你熟练掌握了 GAMBIT, 那么在 CFD 应用中你将如虎添翼。让我们赶紧进入 GAMBIT 的学习吧。

3.1 对连续场的离散化处理

现阶段对非定常（完全）N-S 方程的直接数值求解往往受到计算机运行速度和内存大小的限制尚不现实，而且工程上对瞬时流场也不感兴趣，因此在实际应用中一般是从简化的数学模型出发，并要在简化模型的复杂程度和可处理的几何外形的复杂程度之间作出某种权衡，要求对模型的合适程度和计算的可行性（物理上和几何上）作出判断。目前计算流体力学完全可以模拟具有复杂几何外形的简单物理问题或者模拟具有简单几何外形的复杂物理问题，而不能完全模拟既具有几何复杂性又具有物理复杂性的问题，对此仍在进一步发展中。完全 N-S 方程按时间平均并按从高到低的层次可简化成雷诺平均 N-S 方程、边界层方程、无粘非线性方程（如 Euler 方程、位势方程、跨音速小扰动方程）、无粘线性方程（如 Lap1ace 方程）等。从数值求解上述控制方程的进程来看，20 世纪 60 年代解决了无粘线性方程的求解，已能用无粘线性方程模拟相当复杂外形的小攻角绕流，并有大量的实用软件；20 世纪 70 年代主要集中于无粘非线性全位势方程和 Eu1er 方程的求解，已能用于模拟许多复杂外形的亚、跨、超音速绕流；20 世纪 80 年代较集中于求解雷诺平均 N-S 方程及其它近似的 N-S 方程，着重解决定常问题，已取得了丰硕的成果，并趋于成熟；20 世纪 90 年代开始了非定常粘性流场模拟的新局面，并且它已逐渐成为计算流体力学的发展主流。其中关键性的技术是发展湍流运动的更高层次的数值模拟，主要有：雷诺应力模拟和代数应力模拟、大涡模拟（LES）、直接数值模拟（DNS）。大涡模拟和直接数值模拟的主

要目的是为了更好地理解和认识湍流结构，以便得到更好的湍流模型。

现最多采用的仍是由偏微分方程或积分方程离散化的方法来模拟流场，所以计算方法仍是最重要的研究课题。它不仅直接关系到数值计算的稳定性、收敛性和计算效率，而且直接关系到计算结果的正确性和分辨率。对偏微分方程或积分方程进行离散的方法有多种，最初发展起来的是有限差分法（FDM）,至今其应用仍非常广泛；近年来，人们还越来越广泛的使用有限体积法（FVM）、有限元法（FEM）等。在 GAMBIT 中使用的是有限体积法。

为了模拟可压缩非定常、有分离、有间断（如激波）和有湍流的复杂流动，人们提出了隐式近似因式分解格式、矢通量分裂格式、显式时间多步推进的有限体积法、守恒率系单调上风（MUSCL）格式、总变差减少（TVD）格式、本质无振荡（ENO）格式、守恒同族特征方法（CSCM）、无波动无自由参数的耗散（NND）格式、界值为限（MmB）格式、紧致迎风格式、耗散比拟方法等各种算法，为世人瞩目。而为了提高计算效率，人们发展了隐式残值光顺技术、波吸收型边界条件处理法、远场阻尼法、多重网格法、预处理技术与预处理共轭梯度法（PCG）、谱移动法等。

3.2　网格生成技术

计算流体力学的本质就是对控制方程在所规定的区域上进行点离散（如有限差分法）或区域离散（如有限元法与有限体积法），从而转变为在各网格点或子区域上定义的代数方程组（如果是非线性方程组通常还要进行线性化），然后用线性代数的方法迭代求解。网格生成是其中的一个关键步骤，当物体外形复杂程度较大时，网格生成技术将起到至关重要的作用。网格可分为两大类：结构网格和非结构网格。

3.2.1　结构网格生成技术

过去普遍采用的是结构网格。所谓结构网格就是网格拓扑相当于矩形域内均匀网格的网格。为了便于处理物面边界条件，以提高计算精度，常要求结构网格具有贴体性质，即通过坐标变换，使物体的几何边界成为坐标面（线）。现有的结构网格的生成方法基本上可分为以下四大类：

1、代数生成方法。其特点是根据边界上规定的网格点位置，或者附加一些参考点位置，用插值方法确定所有其它网格点的位置。它具有简便灵活、计算速度快的突出优点，但对复杂的几何形状往往难以找到合适的插值函数。

2、保角变换方法。它能生成完全正交的贴体网格，计算机时也少，但局限于二维情况，且对物体形状往往有很大限制。

3、偏微分方程方法。其特点是通过求解偏微分方程的边值问题来确定区域内网格点分布。它具有较大的适应性，且生成的网格质量很好，特别是椭圆型方程生成的网格通常是光滑和均匀变化的，同时调和函数的极值性质保证了网格生成时物理空间和计算空间之间

的一一对应关系，但网格较密时，一般需要较长的计算机时。

4、变分原理方法。在这类方法中，将生成网格所希望满足的要求表示成某个目标函数（泛函）取极值。这种方法常用于生成自适应网格，因为可以比较方便地将自适应网格的要求用某个变分原理来表示，然后再导出和该变分原理相应的偏微分方程，即 Euler 方程。

采用结构网格总的优点是可以方便准确地处理边界条件，计算精度高，并且可以采用许多高效隐式算法和多重网格法，计算效率也较高。缺点是对复杂外形的网格生成较难，甚至难以实现；即使生成多块结构网格，块与块之间的界面处理又十分复杂，因而在使用上受到限制。

3.2.2 非结构网格生成技术

为了灵活方便地数值模拟绕复杂外形的流动，在 20 世纪 80 年代末人们提出了采用非结构网格的技术手段，现已成为研究的热点之一。所谓非结构网格就是指这种网格单元和节点彼此没有固定的规律可循，其节点分布完全是任意的。其基本思想基于这样的假设：任何空间区域都可以被四面体（三维）或三角形（二维）单元所填满，即任何空间区域都可以被四面体或三角形为单元的网格所划分。它有两种类型：宏观非结构网格和微观非结构网格。宏观非结构网格是先将空间区域划分成许多小块，每个小块用结构网格划分，再将每个小块网格用非结构网格连接，合并成总体网格。它在一定程度上可以解复杂边界问题，但块网格拓扑的形成往往需要大量的人工处理，因而不易形成工程实际所需的通用程序；而且这种网格由于自适应处理会导致网格结构性的彻底破坏，因而也不适合于作网格自适应。而微观非结构网格是完全没有规律的、自由生成的网格，是一种任意的网格。这种网格较结构网格有如下优点：（1）适合于复杂区域的网格划分，特别对奇性点的处理很简单；（2）其随机的数据结构更易于作网格自适应，以便更好地捕获流场的物理特性；（3）其生成过程不需求解任何方程。因而这类网格目前使用较多。其生成主要有两个环节：（1）如何在计算域内合理分布网格点；（2）如何将网格点有效连接，形成三角形或四面体网格单元。现有的生成方法很多，但绝大多数都基于 De1aunay 原理（二维），最为常用的是以下三种：

1、四叉树（二维）/八叉树（三维）方法。该方法的基本思想是先用一个较粗的矩形（二维）/立方体（三维）网格覆盖包含物体的整个计算域，然后按照网格尺度的要求不断细分矩形（立方体），即将一个矩形分为四（八）个子矩形（立方体），最后将各矩形（立方体）划分为三角形（四面体）。例如一个没有边上中间点的矩形可以划分为两个三角形，一个没有棱上中间点的立方体可以划分为五个或六个四面体，对于流场边界附近被边界切割的矩形（立方体），则需考虑各种可能的情况，作特殊的划分。

四叉树/八叉树方法是直接将矩形/立方体划分为三角形/四面体，因此这种方法不涉及邻近点面的查寻，以及邻近单元间的相交性和相容性判断等问题，所以网格生成的速度很快。不足之处是网格质量较差，特别是在流场边界附近，被切割的矩形/立方体的形状可能千奇百怪，由此而划分的三角形/四面体的品质难以保证。尽管如此，四叉树/八叉树作为一

种数据结构已被广泛应用于阵面推进法和 Delaunay 方法中，已提高查寻效率。

2、Delaunay 方法。该法首先是在区域内分布节点，然后采用适当的准则将点连接成四面体单元，生成空间网格。最为常用的准则为球形准则，即一个四面体的外接圆内不包含其它点。该法的优点是方便、速度快，但它需要判断物面边界并进行额外处理，而且如何能保持物面形状也是一个难题。

3、阵面推进方法。该法有三个基本步骤：（1）生成背景网格；（2）生成三维物体的表面网格，形成初始阵面；（3）引入新结点，推进阵面，生成空间网格。背景网格是用户自定义的一个包围所感兴趣区域的空间网格，该网格内对单元设置不同的参数来控制单元的大小和形状。初始推进面是将边界点连接而成的表面网格，边界点由原始数据和背景网格确定，这里要定义初始推进面及其法线方向，确保推进面向区域内部推进。推进面向外推进，在推进过程中生成新的点和四面体单元，并在生成过程中对点、线、面进行插入、删除等工作，直到推进面上所有三角形单元消失，即区域完全被填满为止，网格生成结束。该法的优点是人工干预少，但因空间点是任意给定的，故网格质量较难控制。

实际应用时往往将上述方法结合起来。需要特别指出的是，对于非结构网格技术，仍有一些棘手的问题急待解决：首先，非结构网格的生成，特别是三维情况，是十分耗机时的繁琐工作，寻求通用、方便、快速、有效的生成方法是急需解决的。其次，由于非结构网格的特殊性，高精度的有限差分格式如何与之相结合，也是一个非常重要的问题。迄今，高精度差分格式还主要应用于十分规则的矩形网格，在非正交网格坐标下，未必能得到高精度的解。对于非结构网格而言，具有三阶以上的高精度格式尚难以应用。再次，不能简单将一些基于结构网格的成熟的差分格式和高效隐式算法直接推广应用于非结构网格，比如近似因式分解格式和交替方向隐式（ADI）算法就无法采用。如果在这些方面能取得突破，采用非结构网格技术将如虎添翼。

此外，为了提高解的精度，网格点必须足够密，而整体加密网格所增加的计算量是无法忍受的，因而可以作网格的自适应处理，即使生成的网格可以随求解过程变动，或者在流动参数变化较大的区域能够自动加密。非结构网格的自适应处理很方便，使得自适应网格成为数值计算中提高计算效率和求解精度的一种重要手段。现有的网格自适应方法有再生网格（Remeshing）法、网格加密（Mesh Refinement）法。网格移动（Mesh Movement）法等。再生网格法需要较长的计算机时，而网格移动法又相当复杂，一般采用网格加密法。

3.2.3　混合网格生成技术

如前所述，结构网格和非结构网格各有优缺点，自然就会想到如何将这二者的优势结合起来，同时克服各自的不足，由此混合网格技术应运而生，并越来越受到重视。

将结构网格和非结构网格混合起来的方案很多，其中主要有以下几种。

1、针对多部件或多体复杂外形的混合网格。这类混合网格是先对多体问题的每一单体或复杂外形的每一部件生成贴体结构网格，而在体与体、部件与部件之间的交界区挖出一

个洞，洞内由非结构网格来填充。这类混合网格的代表有“拉链”（Zipper Grids）和“龙型”网格（DRAGON Grids）等。

2、针对黏性计算的混合网格。这类混合网格是先在物面附近向外推出数层有一定压缩比的结构网格（二维）或半结构网格（三维、三棱柱）以模拟边界层，然后外场用非结构网格。事实上，在外形不太复杂的情况下，全场采用三棱柱网格也是可行的。

3、矩形与非结构混合网格。矩形网格中不必进行 Jacobian 矩阵计算，具有比贴体网格更为简单更为快捷的优点。但其不足之处在于不易处理曲面边界，处理得不好就会出现所谓“台阶效应”。事实上，在物面附近采用非结构网格就可以消除“台阶效应”，同时达到模拟复杂外形的目的。这就形成了矩形/非结构混合网格。为了计算黏性问题，亦可在物面附近采用结构（二维）或半结构（三维）网格，然后由非结构网格过渡到外场的矩形网格，由此构成矩形/非结构/半结构混合网格。

3.3　复杂外形网格生成

通常很难在单块结构网格上计算复杂外形绕流，一般采用以下方法：

l、多块结构网格方法。它将原始的物理求解域按不同的空间网格拓扑分成若干个子区域，每个子区域上网格拓扑简单，易于生成结构贴体网格，然后将每个块组合起来，各块交界面处的网格线可以穿透也可以不穿透。如果网格线穿透，为计算精确，应尽量使网格线光滑地过渡；如果网格线不穿透，即互相交错，此时应特别注意交接面处流动信息的准确传递。该法的优点是：可以使用结构网格上成熟的高效算法；块与块之间的数值守恒性容易保证；交接面处流动信息的传递不需要插值解向量。缺点是：自动空间分块较难，通常需要人工参与；程序编制复杂，必须输入各块联系的拓扑关系。

2、Chimera 网格方法。它采用一种特殊的网格嵌入技术，其特点是：（1）可在整个计算域中任意形成所需的若干子域；（2）各子域间的关系具有层次结构和彼此覆盖的特点；（3）覆盖域内各子域网格之间的通讯采用插值方法实现；（4）可在原有网格系上任意增加一个子域而无需变动原有网格系；（5）一个网格子域可相对于其它网格域自由移动。在应用 Chimera 网格系时，首先独立地生成各子域的结构网格，然后搜索重叠区、标志边界点和内插点等，将各子域组合起来。该法的优点是：网格生成简单，基本可以自动完成，可以保留高效算法。难点是：要依靠解向量的插值来传递各子域网格之间的流动信息，因而很难保证总体守恒性。在求解含有激波的流动问题时是一个困难，对此已做了一些工作来克服，比如改进的全守恒性交接面处理格式。

3、混合网格方法，即采用结构网格和非结构网格混合的方法来求解。其优点是：（1）全守恒；（2）由于流场中大部分区域是结构网格，只有很小部分是非结构网格，因而同样可在结构网格上使用高效算法，以保证整个流场求解的高效率；（3）通常结构网格位于物面附近，非结构网格仅存在于结构网格的连接处，因而可以得到光滑的物面网格分布，宜于求解复杂外形的粘性绕流。

4、非结构网格方法。由于非结构网格非常容易处理具有复杂几何边界的问题，且很容

易作网格自适应，因而该法正被广泛地应用于复杂外形的绕流计算中。但现在大多数非结构网格数值解法都是建立在有限元法或有限体积法之上，主要应用于可压缩流动计算。对于不可压缩流动计算的应用还比较少，主要原因是不可压缩流动的控制方程中没有压力对时间的偏导数项，压力耦合的求解比较困难。解决此困难有不同的方法，如拟压缩性方法，它使得控制方程与可压缩流动的控制方程相似，从而使求解得以简便。

3.4　用 GAMBIT 生成网格的步骤

Gambit 软件提供了功能强大、灵活易用的网格划分工具，可以划分出满足 CFD 特殊需要的网格。

3.4.1　生成线网格

在线上生成网格，作为将在面上划分网格的网格种子，允许用户详细的控制在线上节点的分布规律，Gambit 提供了满足 CFD 计算特殊需要的五种预定义的节点分布规律。

3.4.2　生成面网格

对于平面及轴对称流动问题，只需要生成面网格。对于三维问题，也可以先划分面网格，作为进一步划分体网格的网格的网格种子。

Gambit 根据几何形状及 CFD 计算的需要提供了三种不同的网格划分方法:

1、映射方法

映射网格划分技术是一种传统的网格划分技术，它仅适合于逻辑形状为四边形或三角形的面，它允许用户详细控制网格的生成。在几何形状不太复杂的情况下，可以生成高质量的结构化网格。

2、子映射方法

为了提高结构化网格生成效率，Gambit 软件使用子映射网格划分技术。也就是说，当用户提供的几何外形过于复杂，子影射网格划分方法可以自动对几何对象进行再分割，使在原本不能生成结构化网格的几何实体上划分出结构化网格。子映射网格技术是 FLUENT 公司独创的一种新方法，它对几何体的分割只是在网格划分算法里进行，并不真正对用户提供的几何外形做实际操作。

3、自由网格

对于拓扑形状较为复杂的面，可以生成自由网格，用户可以选择合适的网格类型(三角形或四边)。

3.4.3　边界层网格

CFD 计算对计算网格有特殊的要求，一是考虑到近壁粘性效应采用较密的贴体网格，二是网格的疏密程度与流场参数的变化梯度大体一致。

对于面网格，可以设置平行于给定边的边界层网格，可以指定第二层与第一层的间距比，及总的层数。

对于体网格，也可以设置垂直于壁面方向的边界层，从而可以划分出高质量的贴体网格。而其它通用的 CAE 前处理器主要是根据结构强度分析的需要而设计的，在结构分析中不存在边界层问题，因而采用这种工具生成的网格难以满足 CFD 计算要求，而 Gambit 软件解决了这个特殊要求。

3.4.4　生成体网格

对于三维流动问题，必须生成三维实体网格。Gambit 提供五种体网格的生成方法。

1、映射网格

对于六面体结构，可以使用映射网格方法直接生成六面体网格。对于较为复杂的几何形体，必须在划分网格前将其分割为若干格六面体结构。

2、子映射网格

Gambit 软件的子映射网格划分技术同样适用于体网格。也就是说，当用户提供的几何外形过于复杂，子影射网格划分方法可以自动对几何对象进行再分割，使在原本不能生成结构化网格的几何实体上划分出结构化网格。

3、Cooper 方法

Cooper 方法适合于在一个方向几何相似，而在另两个方向几何较为复杂的实体。

4、Tgrid 方法

对于复杂的工程结构，可以采用 Tgrid 方法方法生成四面体和金字塔网格，Tgrid 方法生成网格过程不需要用户干预，可以划分出网格密度变化很大的网格。特别适合计算域很大的外流场。

5、混合网格

对于复杂的工程结构，可以综合使用所由的网格生成方法。在贴近壁面处可以生成结构化网格，在不需要严格控制的地方，可用 Tgrid 方法生成自由网格。Gambit 软件可以根据几何结构特点和现有的网格约束条件迅速生成网格。混合网格技术的应用将大大减少网格划分时间，同时又能保证网格的质量。

3.5 GAMBIT 的图形用户界面

网格的划分使用 Gambit 软件，首先要启动 Gambit，在 Dos 下输入 Gambit <filemane>，文件名如果已经存在，要加上参数-old。Gambit 的图形用户界面如下图所示：

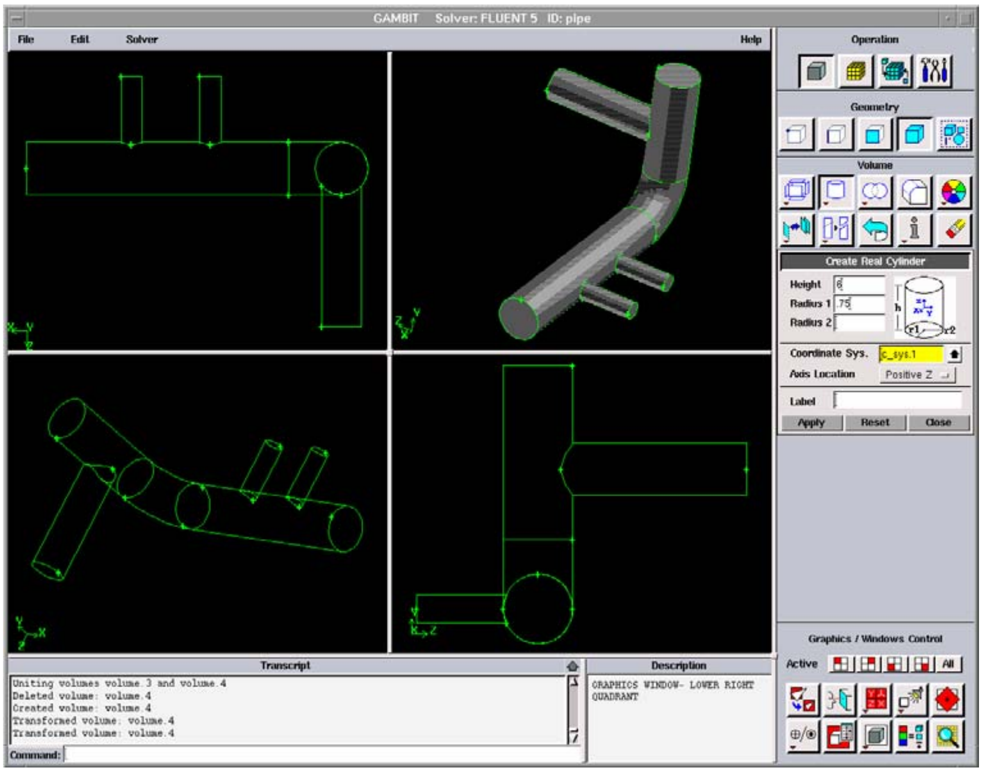


图 3.5.1 Gambit 操作界面

如图 3.5.1 所示，Gambit 用户界面可分为 7 个部分，分别为：菜单栏、视图、命令面板、命令显示窗、命令解释窗、命令输入窗和视图控制面板。

菜单栏

菜单栏位于操作界面的上方，其最常用的是 File 命令下的 New、Open、Save、Save as 和 Export 等命令。这些命令的使用和标准的软件一样。Gambit 可识别的文件后缀为.dbs，而要将 Gambit 中建立的网格模型调入 Fluent 使用，则需要将其输出为.msh 文件(file/export)。

视图和视图控制面板

Gambit 中可显示四个视图，以便于建立三维模型。同时我们也可以只显示一个视图。视图的坐标轴由视图控制面板来决定。图 3.5.2 显示的是视图控制面板。

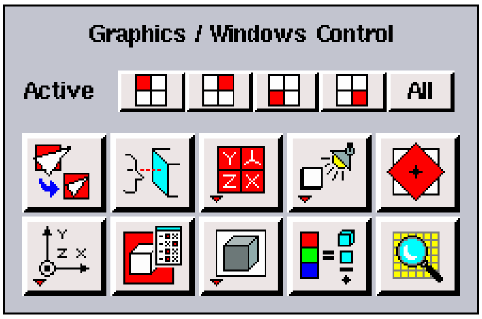
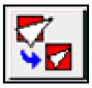
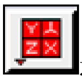
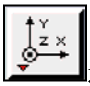
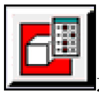
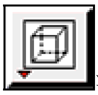


图 3.5.2　视图控制面板

视图控制面板中的命令可分为两个部分，上面的一排四个图标表示的是四个视图，当激活视图图标时，视图控制面板中下方十个命令才会作用于该视图。

视图控制面板中常用的命令有：

 全图显示、选择显示视图、选择视图坐标、选择显示项目、渲染方式。

同时，我们还可以使用鼠标来控制视图中的模型显示。其中按住左键拖曳鼠标可以旋转视图，按住中键拖动鼠标则可以在视图中移动物体，按住右键上下拖动鼠标可以缩放视图中的物体。

3.6　GAMBIT 菜单命令

命令面板是 Gambit 的核心部分，通过命令面板上的命令图标，我们可以完成绝大部分 Gambit 的使命。

图 3.6.1 显示的就是 Gambit 的命令面板。

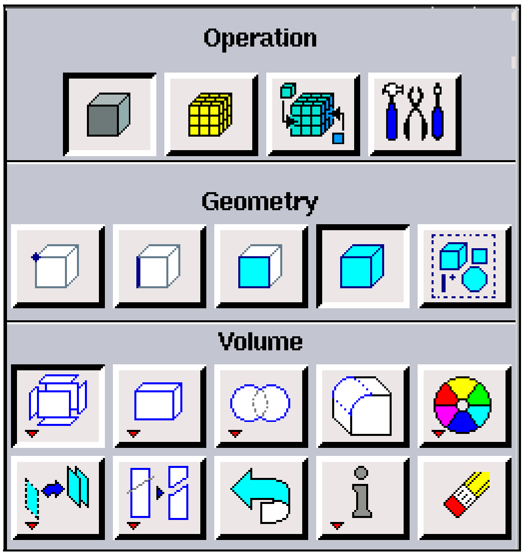


图 3.6.1 Gambit 的命令面板

从命令面板中我们就可以看出，网格生成的工作可分为三个步骤：一是建立模型，二是划分网格，三是定义边界。这三个部分分别对应着 Operation 区域中的前三个命令按钮 Geometry(几何体)、mesh（网格）和 Zones（区域)。Operation 中的第四个命令按钮 Tools 则是用来定义视图中的坐标系统，一般取默认值。命令面板中的各个按钮的含义和使用方法将在以后的具体例子中介绍。

命令显示窗和命令输入栏

命令显示窗和命令输入栏位于 Gambit 的左下方（如图 3.6.2 所示)。

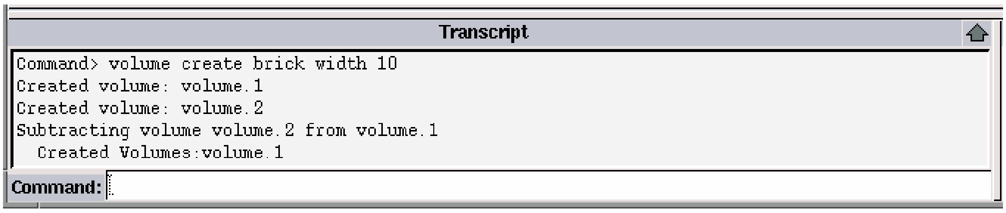


图 3.6.2 命令显示窗和命令输入栏

命令显示窗中记录了每一步操作的命令和结果，而命令输入栏则可以直接输入命令，其效果和单击命令按钮一样。

命令解释窗

图 3.6.3 显示的是位于命令显示窗左方的命令解释窗 ，当我们将鼠标放在命令面板中任意一个按钮的上面，Description 窗口中将出现对该命令的解释。

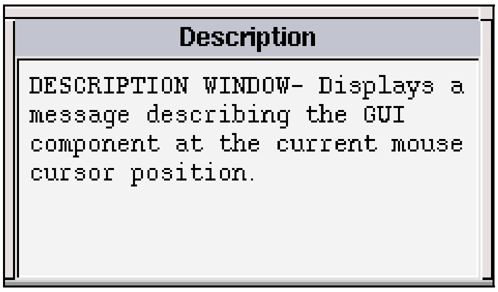


图 3.6.3　命令解释窗

3.7　用 GAMBIT 创建基本二维几何模型

Gambit 创建几何模型的功能还是较为强大的。划分网格的第一步就是要建立模型。在命令面板中单击 Geometry 按钮，进入几何体面板。

图 6 显示了几何体面板中的命令按钮。

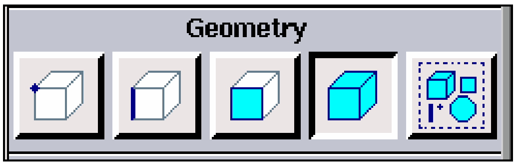


图 3.7.1 命令按钮

图 3.7.1 中从左往右依次是创建点、线、面、体和组的命令。

对于二维模型的建立，一般要遵循从点到线，再从线到面的原则。

以二维轴对称单孔喷嘴的模型创建为例介绍二维模型的生成。]

首先要确定问题的计算域。

计算域的确立

图 3.7.2 是一个二维轴对称单孔喷嘴射流问题的计算区域。由于 Fulent 的边界提法比较粗糙，多为一类边界条件，因此建议在确定计算域时，可以适当加大计算范围。从图中我们可以看出，计算区域为 4D*12D，其中在喷嘴的左边取了 2D 的计算区域，就是为了减小边界条件对计算的影响。



图 3.7.2　计算域的确定

对于上述的计算域，我们在建立计算模型时按照点、线、面的顺序来进行。

创建点（vertex）

单击命令面板中的 Vertex 按钮，进入 Vertex 面板（见图 3.7.3）

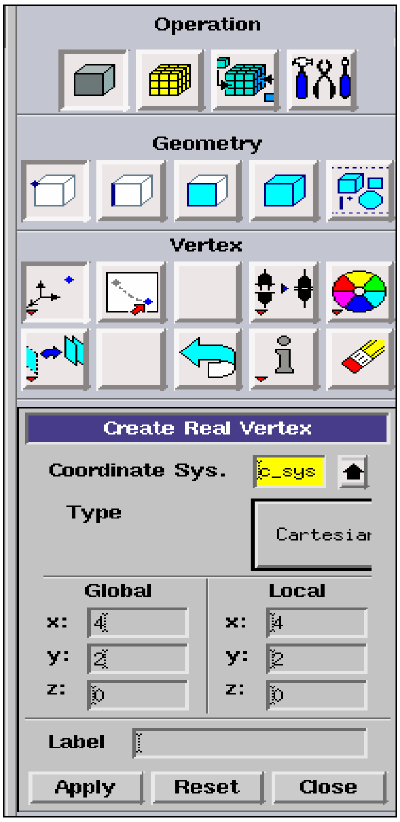


图 3.7.3　Vertex 命令面板

单击 Vertex Create 按钮，在 Create Real Vertex 对话框中输入点的坐标，再单击 Apply 按钮，就可以创建点。计算出计算域的各个顶点的坐标，依次创建这些顶点（见图 3.7.4）。



图 3.7.4　点的创建

在 Gambit 中点的创建方式有四种：根据坐标创建、在线上创建、在面上创建和在体上创建。我们可以根据不同的需要来选择不同的创建方式（见图 3.7.5）。

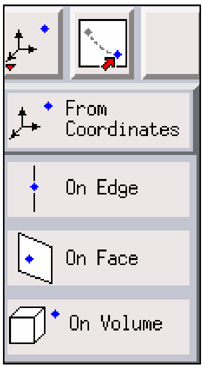


图 3.7.5

Vertex 中常用的命令还有：Move/Copy、Undo 和 Del。

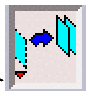
- Move/Copy 命令 

图 3.7.6 显示的是 Move/Copy Vertex 对话框。

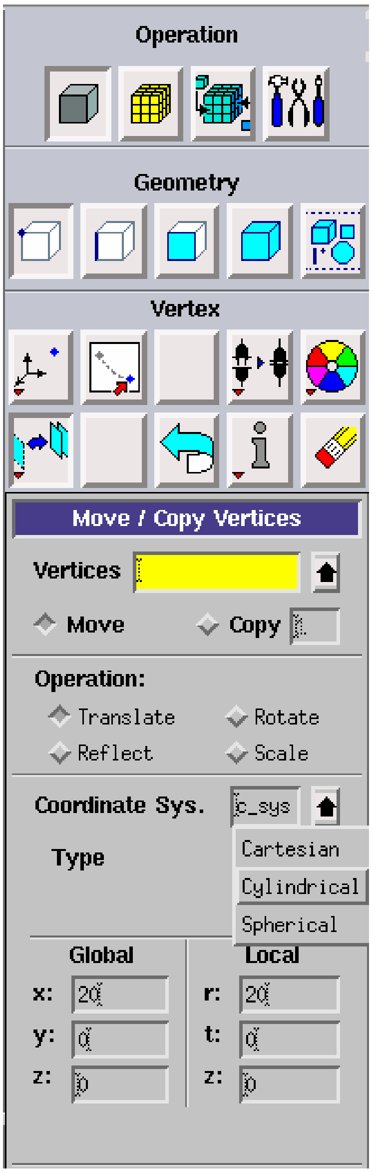


图 3.7.6

当我们要复制或移动一个点时，首先要选择需要作用的点。在命令面板中单击 Vertices 右边的输入栏，输入栏以高亮黄色显示，表明可以选择需要的点。

在 Gambit 中选择一个对象的方法有两种：

1. 按住 Shift 键，用鼠标左键单击选择的对象，该对象被选中，以红色显示。
2. 单击输入栏右方的向上箭头，就会出现一个对话框，从对话框中可以选择需要的点的名称（见图 3.7.7）。因此为了便于记忆，建议在创建对象的时候要起一个便于记住的名字。

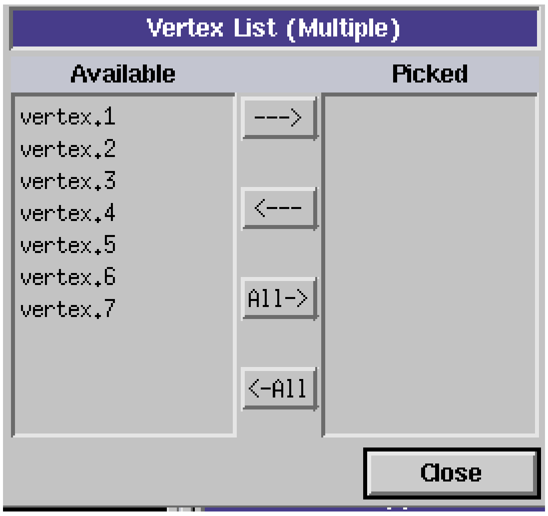
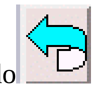
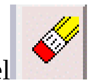


图 3.7.7

同时，Gambit 还为我们提供了三种不同的坐标系，即直角坐标系、柱坐标和球坐标。在命令面板的坐标类型中，可以选择不同的坐标系。

- Undo 

Undo 命令可以消除上一步操作的内容，但需要注意的是，在 Gambit 中只有 Undo 命令而没有 Redo 命令。

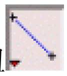
- Del 

Del 命令用来删除一些误操作或不需要的对象。单击 Del 按钮，在视图中选择需要删除的对象，再单击 Apply 按钮即可。

线的创建（Line）

在命令面板中单击 Edge 按钮，就可以进行线的创建和编辑（见图 3.7.8）。

在 Gambit 中，最常用的是直线的创建。

在 Edge 命令面板中单击 Create Straight Edge 按钮，在视图中选择需要连成线的点，单击 Apply 按钮即可（见图 3.7.9）。这时视图中的线段是以黄色显示。当这些线段组成一个面时，将以蓝色显示。

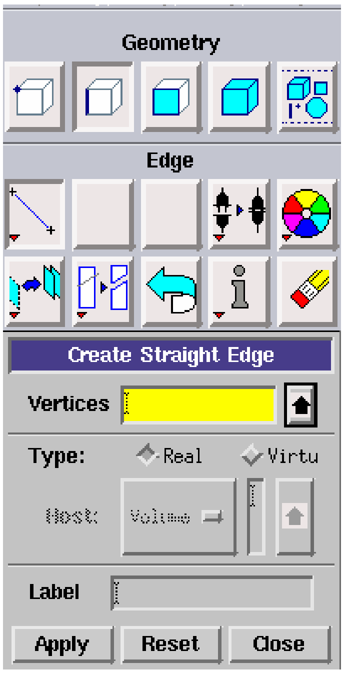


图 3.7.8

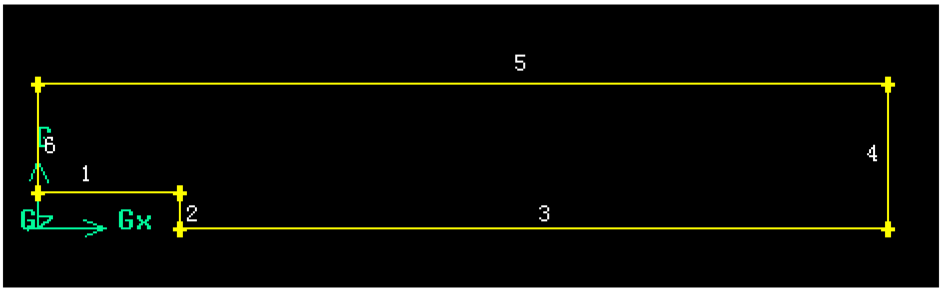


图 3.7.9

除了创建直线外，Gambit 还可以创建其他的一些线段，如圆弧、圆、倒角、椭圆等（见图 3.7.10）

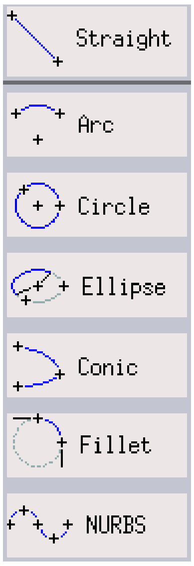
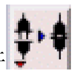
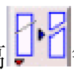


图 3.7.10

Edge 命令中常用的还有合并、分离等命令，即可以把两条线段合成一条，也可以将一条线段分成两条，这些可以为面的创建和网格划分提供方便。因为面的创建需要一个封闭的曲面。

面（Face）的创建

面的创建工作十分简单，只须选择组成该面的线，单击 Apply 按钮即可（见图 3.7.11）。需要注意的是这些线必须是封闭的，同时我们要创建一个二维的网格模型，就必须创建一个面，只有线是不行的。同样的道理，在创建三维的网格模型的时候，就必须创建体。

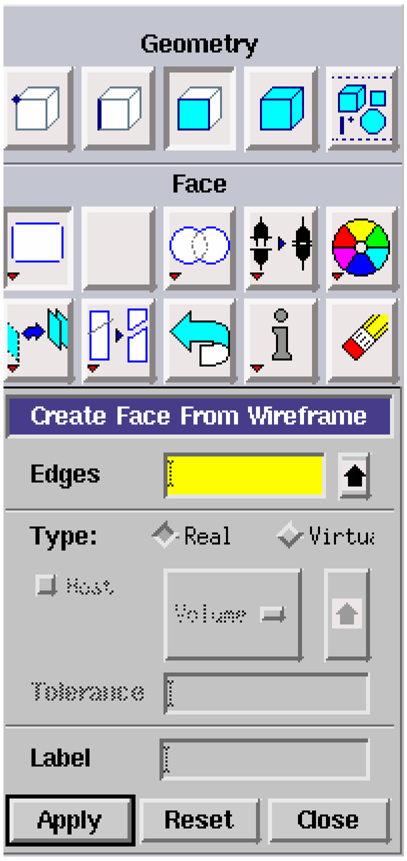


图 3.7.11

在面的创建中，有一个布尔运算的操作，可以使我们创建不规则形状的面（见图 3.7.12）。布尔运算包括三种方式：加、减、交。

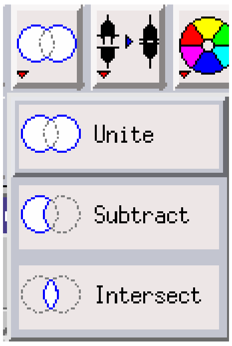


图 3.7.12

3.8　二维网格划分

对于二维网格的建立，一般要遵循从点到线，再从线到面的原则。

以二维轴对称单孔喷嘴的网格划分为例介绍二维网格的生成。

在命令面板中单击 Mesh 按钮，就可以进入网格划分命令面板。在 Gambit 中，我们可以分别针对边界层、边、面、体和组划分网格。图 3.8.1 所示的五个按钮分别对应着这五个命令。

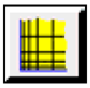
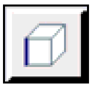
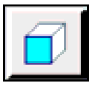
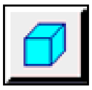
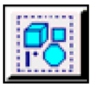
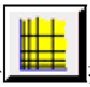
	Boundary Layer （边界层）
	Edge （边）
	Face （面）
	Volume （体）
	Group （组）

图 3.8.1

3.8.1 边界层网格的创建

在命令面板中单击按钮，即可进入边界层网格创建（见图 3.8.2）。

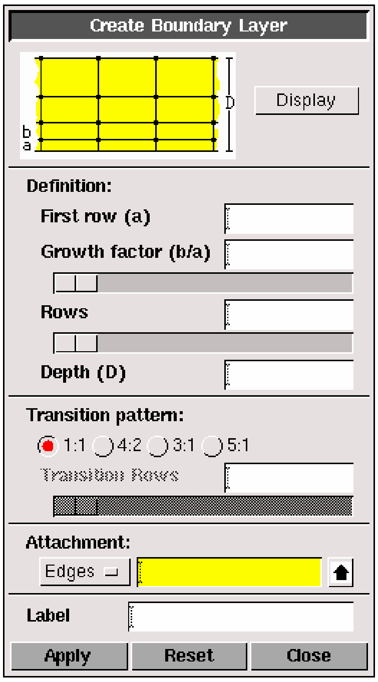


图 3.8.2

边界层网格的创建需要输入四组参数，分别是第一个网格点距边界的距离（First Row），网格的比例因子（Growth Factor），边界层网格点数（Rows，垂直边界方向）以及边界层厚度（Depth）。这四个参数中只要任意输入三组参数值即可创建边界层网格。

同时，我们还可以选择边界层网格创建的形式。在命令面板的 Transition Pattern 区域，系统给我们提供了四种创建方式（见图 3.8.3）。

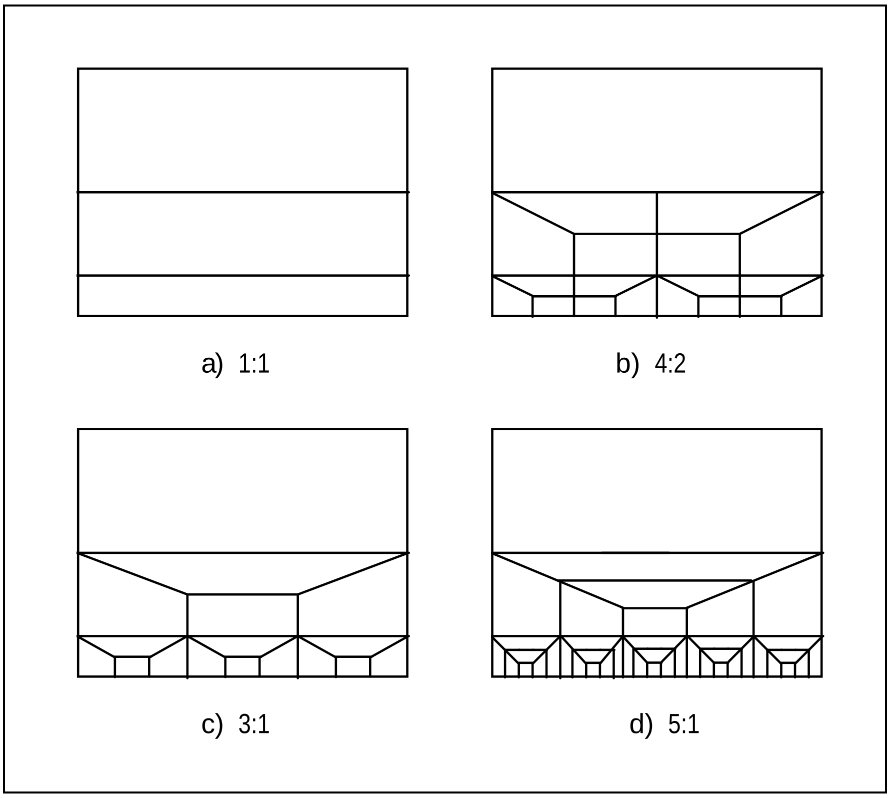


图 3.8.3

以上述二维轴对称圆孔射流的计算模型为例，介绍边界层网格的生成。

1． 单击 Mesh 按钮，选择 Boundary layer 选项，进入边界层网格创建命令面板。

2． 按住 Shift 按钮，用鼠标左键单击图形中的线段 1，选择其为创建对象。

3． 输入参数值为：First Row：0.05，Growth Factor：1.01，Rows：10，选择创建形式为 1:1，单击 Apply 按钮完成创建工作（见图 3.8.4）。

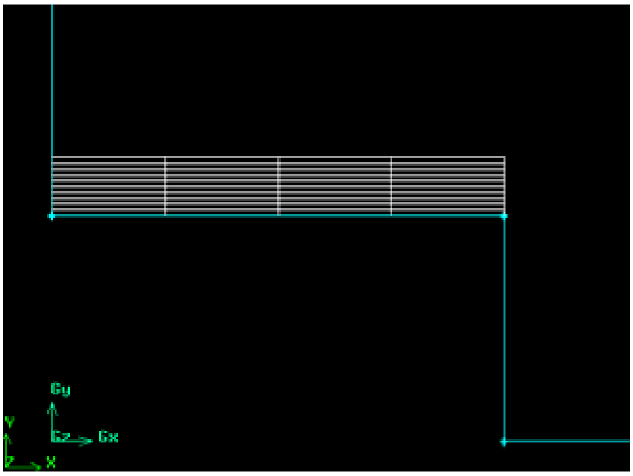
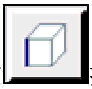


图 3.8.4

3.8.2 创建边上的网格点数

当我们划分的网格需要在局部加密或者划分不均匀网格时，我们首先要定义边上的网格点的数目和分布情况。

边上的网格点的分布可分为两种情况，一种是单调递增或单调递减，一种是中间密（疏）两边疏（密）。下面依然结合实例介绍边上网格点的创建。

1. 单击命令面板中的按钮，进入 Edge 网格创建面板（见图 3.8.5）。

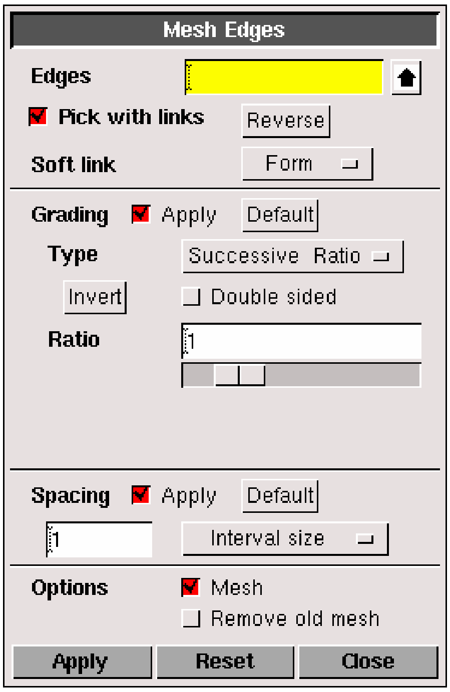


图 3.8.5

2. 在图四中选择线段 2。
3. 在命令面板中单击 Double Side 按钮，设置 Radio1 和 Radio2 为 1.05。
4. 在命令面板中单击 Interval Size 按钮，选择 Interval Count 选项。
5. 在 Interval Count 按钮的左边输入参数值为 20。
6. 单击 Apply 按钮，观察视图中边上的网格点的生成（见图 3.8.6）。

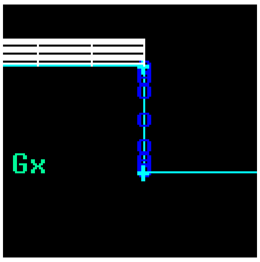


图 3.8.6

7. 选择视图中的线段 3，取消对 Double Side 按钮的选择，设置 Radio 为 1.01，Interval Count 为 80，观察视图中网格点的分布情况。视图中选中线段上的红色箭头代表了 Edge 上网格点分布的变化趋势。如果 Radio 大于 1，则沿箭头方向网格点的分布变疏，小于 1，则沿箭头方向网格点的分布变密。如果发现网格点的分布情况与预计的相反，可以采用两种方法解决：（1）按住 Shift 按钮，在所选择的线段上单击鼠标中键改变箭头的方向；（2）在命令面板中单击 Invert 按钮，将 Radio 值变为其倒数值。

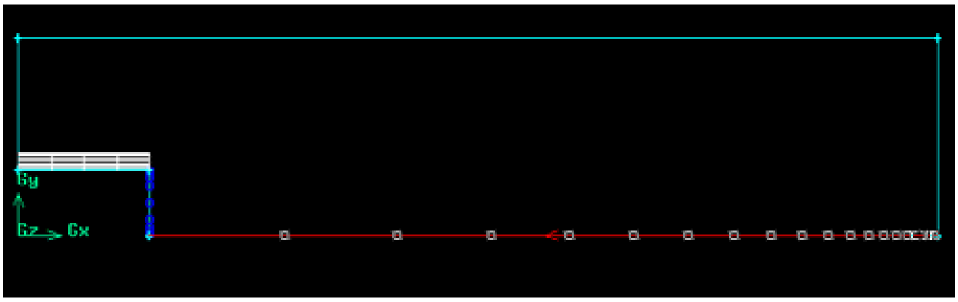


图 3.8.7

8. 依次选择视图中的线段 4、5、6、1，设置合理的网格点分布。

注意：在设置网格点分布的时候，一个封闭面的最后一条线段的网格点的分布可以通过系统自动计算得到。

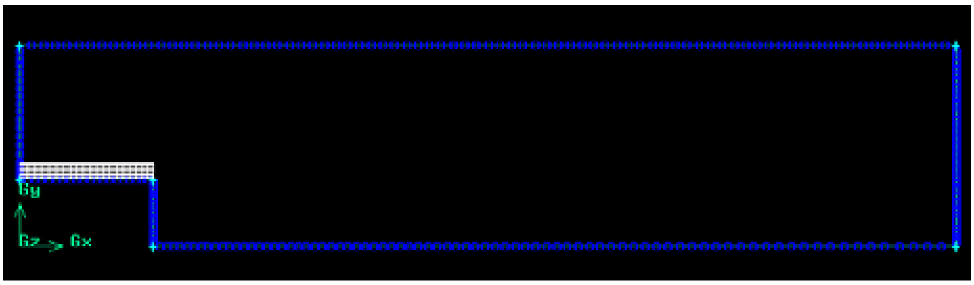


图 3.8.8

3.8.3 划分面的网格

Gambit 对于二维面的网格的划分提供了三种网格类型：四边形、三角形和四边形/三角形混合，同时还提供了五种网格划分的方法。表 1、2 分别列举了五种网格划分的方法以及它们的适用类型。

方法	描述
Map	创建四边形的结构性网格
Submap	将一个不规则的区域划分为几个规则区域并分别划分结构性网格。
Pave	创建非结构性网格
Tri Primitive	将一个三角形区域划分为三个四边形区域并划分规则网格。
Wedge Primitive	在一个楔形的尖端划分三角形网格，沿着楔形向外辐射，划分四边形网格。

表 1

	适用类型		
方法	Quad	Tri	Quad/Tri
Map	×		×
Submap	×		
Pave	×	×	×
Tri Primitive	×		
Wedge Primitive			×

表 2

下面仍然以二维轴对称自由射流的网格划分为例，来介绍各种网格的生成。

1.　单击命令面板中的按钮（Mesh Face），进入面的网格创建命令面板（见图 3.8.9）。

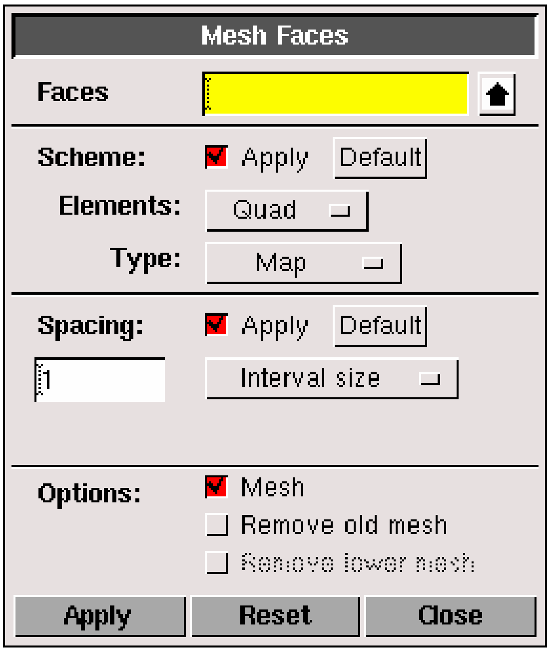


图 3.8.9

2.　选择视图中的面，系统中默认的网格点的类型为四边形结构网格。单击 Apply 按钮，观察网格的生成（见图 3.8.10）。

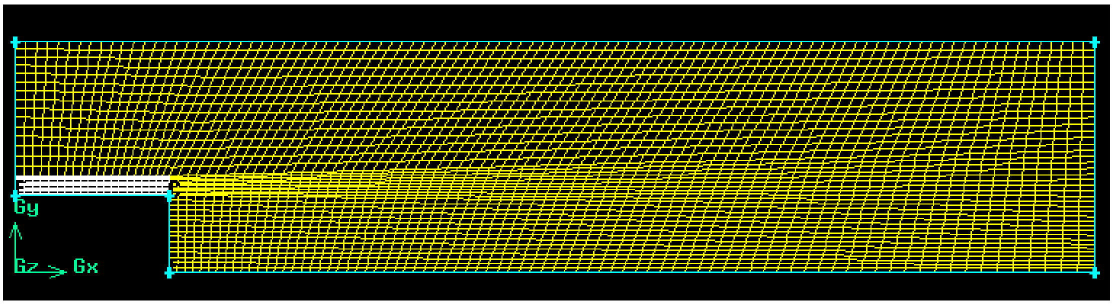


图 3.8.10

3.　在命令面板的 Type 中选择网格类型为 Pave，单击 Apply 按钮，观察网格的生成（见图 3.8.11）。

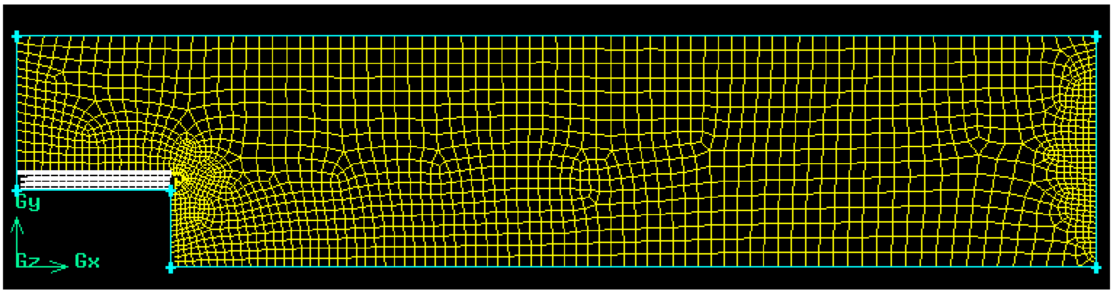


图 3.8.11

4． 选择 Element 类型为 Tri，单击 Apply 按钮，观察网格的生成（见图 3.8.12）。

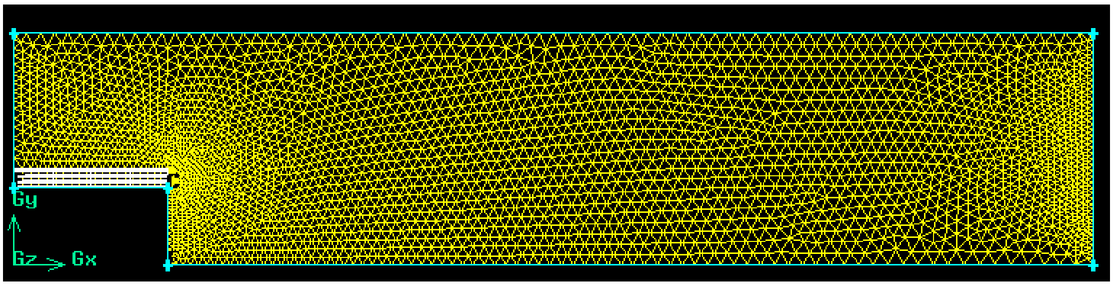
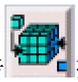


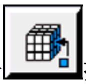
图 3.8.12

3.9 定义二维网格区域类型

在 Gambit 中，我们可以先定义好各个边界条件的类型，具体的边界条件取值在 Fluent 中确定。

1． 在菜单栏中选择 Fluent/Fluent5。这个步骤是不可缺少的，它相当于给 Gambit 定义了一个环境变量，设置完之后，定义的边界条件类型和 Fluent5 中的边界类型相对应。

2． 在命令面板中单击按钮，进入区域类型（Zone Type）定义面板。

3． 单击按钮，出现 Specify Boundary type 对话框（见图 3.9.1）。

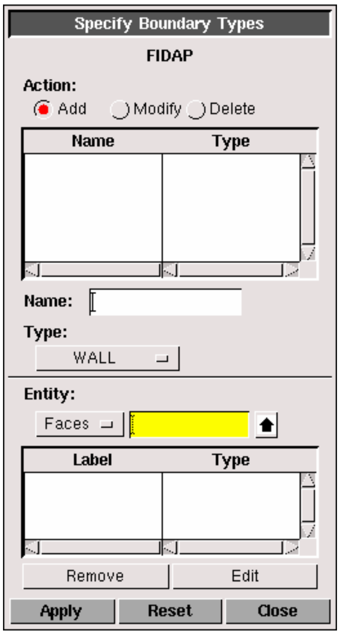


图 3.9.1

4. 选择 Entity 类型为 Edge。在视图中选择 Edge1，在 Name 区域中输入 Wall，选择 Type 为 Wall，即定义 Edge1 的边界条件为固壁条件，取名为 Wall。

5. 选择 Edge2，定义边界条件为压力入流条件（Pressure Inlet），取名为 Inflow。

6. 选择 Edge4，定义边界条件为压力出流条件（Pressure Outlet），取名为 Outflow。

7. 选择 Edge5、6，定义边界条件为远场压力条件（Pressure Far-field），取名为 Outflow1。

选择 Edge3，定义边界条件为轴对称条件（Axis），取名为 Axis。

3.10 网格文件保存和输出

当你完成了网格的划分并定义好网格区域类型之后，你就可以导出网格了。

1. 在菜单栏中选择 File/Save as，在对话框中输入文件的路径和名称。(注意：在 Gambit 中要往一个文本框中输入文字或数字，必须先将鼠标在文本框中单击选中文本框）

2. 选择 File/Export/Mesh，输入文件的路径和名称。

如下面的界面：

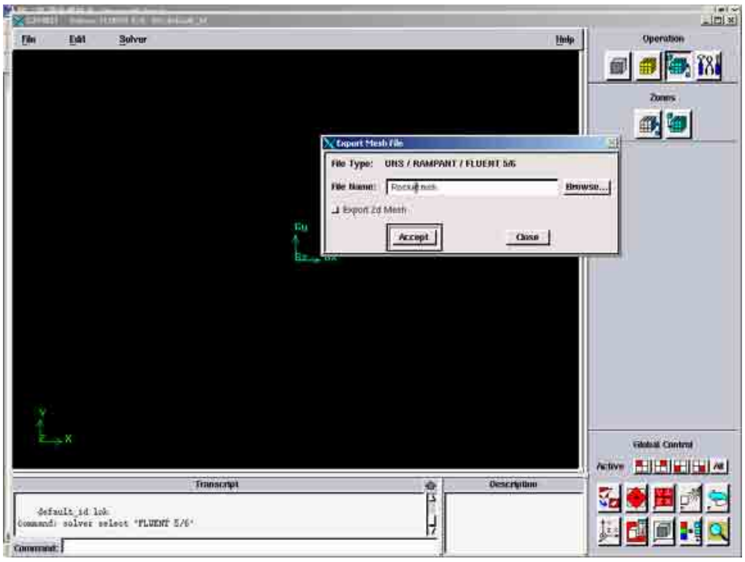


图 3.10.1

然后你填入一个网格文件名或接受默认的文件名，点击 Accept 即完成了导出网格的工作。需要提醒读者的是请您注意下面信息提示窗口的信息，如果是“Done.”的话，那我就要祝贺你了！当然作为一个熟练的使用者，您最好看一下详细的导出信息，其中的每个警告信息都是不可忽视的，因为它会影响你在 Fluent 中的计算，甚至导致网格不能正确导入 Fluent。

3.11　三维建模

相对于二维建模而言，三维建模与二维建模的思路有着较大的区别。二维建模主要遵循点、线、面的原则，而三维建模则更象搭积木一样，由不同的三维基本造型拼凑而成，因此在建模的过程中更多的用到了布尔运算及 Autocad 等其他的建模辅助工具。

3.11.1　三视图的使用

在建立三维图形的时候，使用三视图有利于我们更好的理解图形。

图 3.11.1 显示的是 Gambit 的视图控制面板。

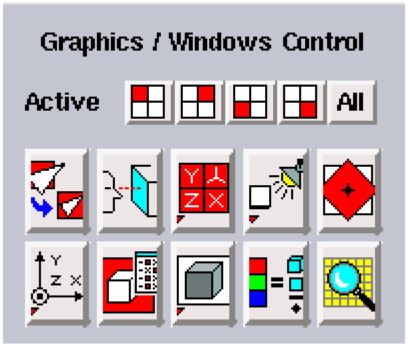


图 3.11.1

在当前状况下，四个视图都是激活的（在 Active 栏中，显示红色），这时视图控制面板中的十个命令将同时作用于四个视图。

在创建三维图形之前，我们要做的第一项工作就是要将 Gambit 的四个视图设置为顶视图、前视图、左视图和透视图。

1. 用鼠标单击 Active 右边的后三个视图，取消对它们的激活，激活取消后呈灰色（见图 3.10.2）。

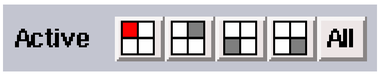
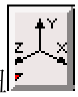
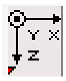
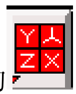
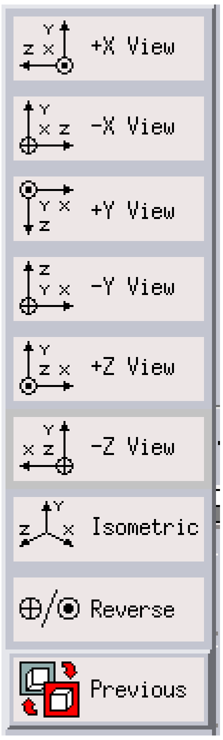


图 3.11.2

2. 用鼠标右键单击视图控制面板中的坐标按钮，弹出一组坐标系（见图 3.11.3）。

3. 选择，则左上视图变成顶视图。如法炮制，设置其他视图（见图 3.11.4）。

4. 单击控制面板中的，也可将视图设成三视图。


+X View
-X View
+Y View
-Y View
+Z View
-Z View
Isometric
Reverse
Previous
图 3.11.3

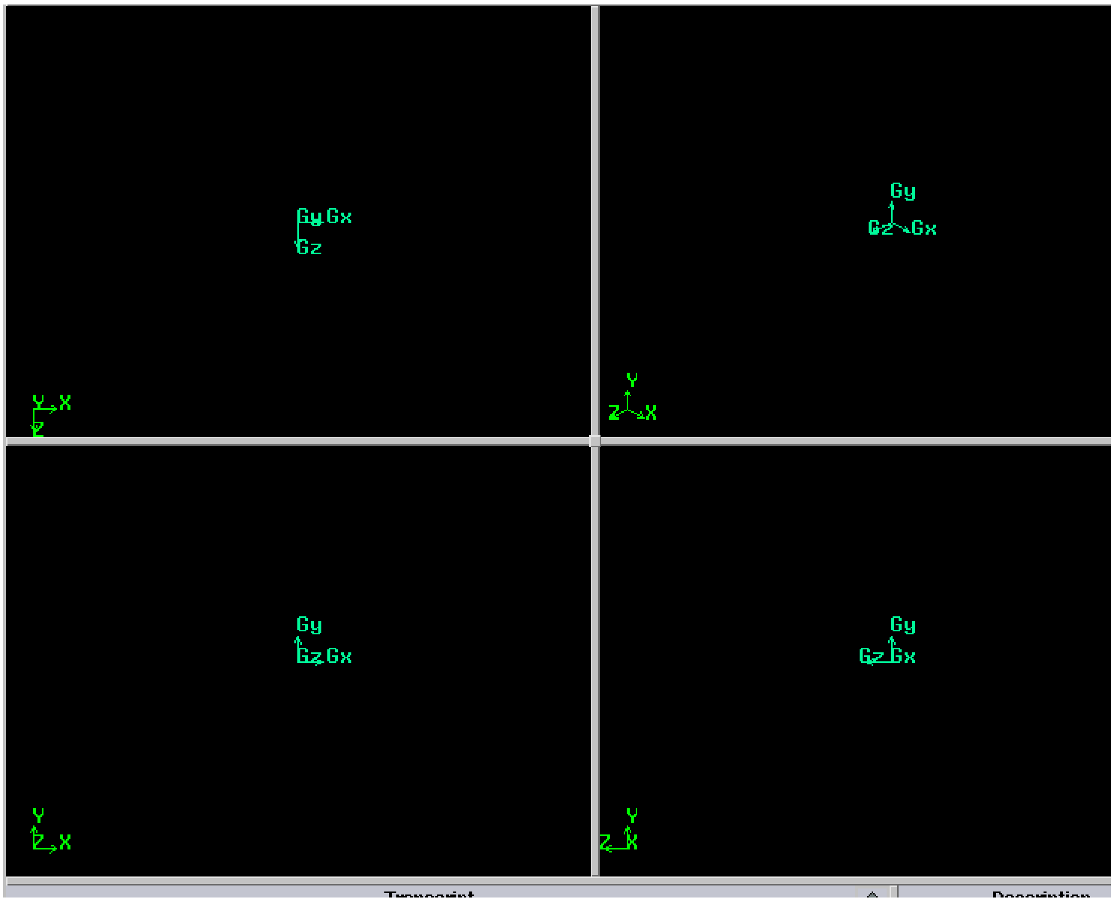
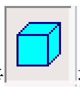
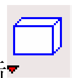


图 3.11.4

3.11.2 基本三维模型的建立

在 Gambit 控制面板中单击按钮，在 Volume 中用鼠标右键单击，弹出一组按钮（见图 3.11.5），表示 Gambit 所能创建的基本三维几何体，主要有长方体、圆柱体等。

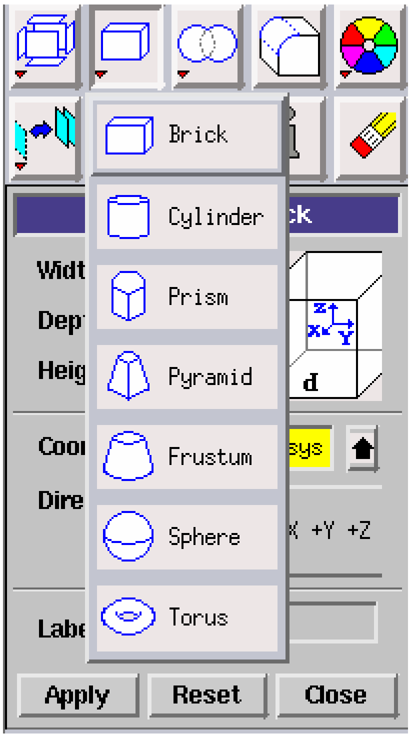
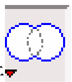
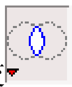


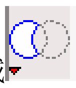
图 3.11.5

布尔运算的基本概念

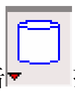
典型的布尔运算包括并、交、减。

并：将两个物体并成一个物体（两个物体的并集）

交：两个物体的交集

减：A 物体减去 B 物体

下面用一个简单的例子来说明基本三维几何体的创建和布尔运算的运用

1．单击按钮，输入参数创建一个高 60，半径 6 的圆柱体（见图 3.11.6）。在 Axial Loaction 栏中选取 Positive X，使得圆柱体的法线指向 x 方向。在 Gambit 中创建的几何体，其基点都在坐标系的原点（见图 3.11.7）。如果创建的几何体过大，在视图中无法显示全

图，或者太小，无法分辨，单击按钮即可。

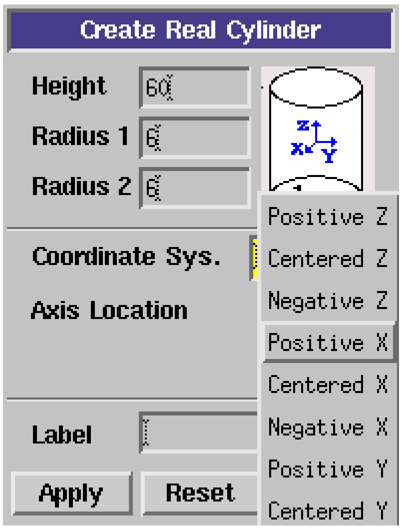


图 3.11.6

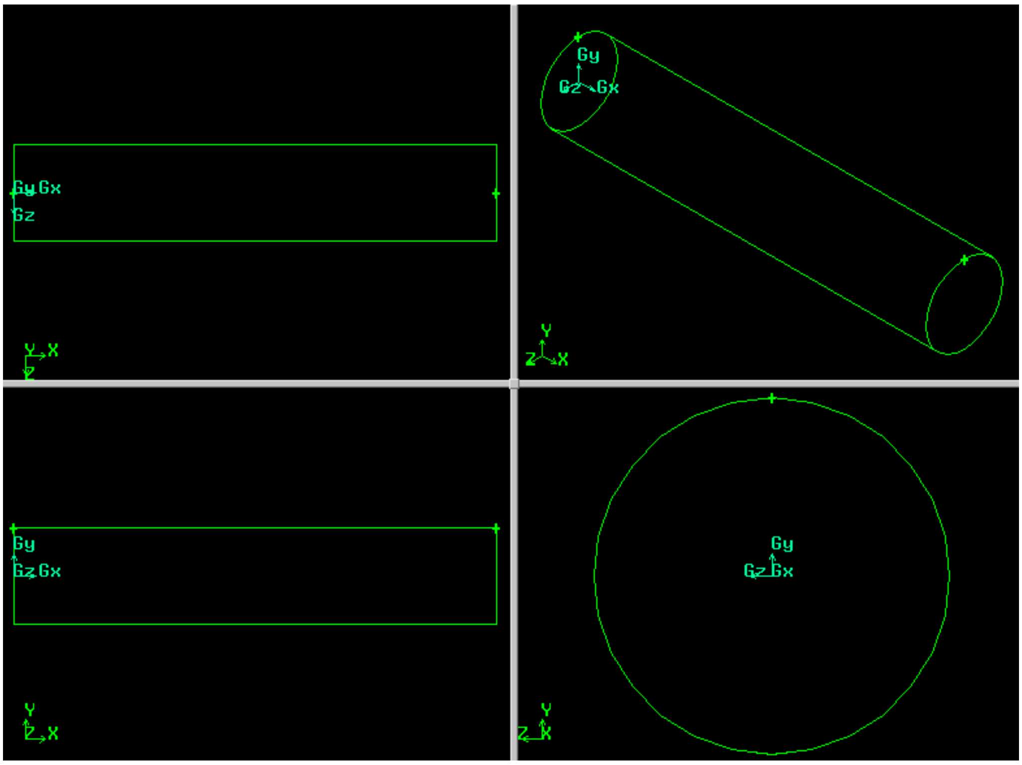
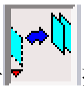


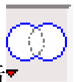
图 3.11.7

2．为了能够更好的观察三维几何体，可以用鼠标拖动四个视图中央的小方块，改变四

个视图的大小（见图 3.11.8）。

3．再创建两个圆柱体，分别指向 y 和 z 方向（见图 3.11.9）。

4．单击按钮，移动圆柱体，使其如图 3.11.10 所示。

5．单击按钮，选择三个圆柱体，依次将它们合并在一起（见图 3.11.11）。

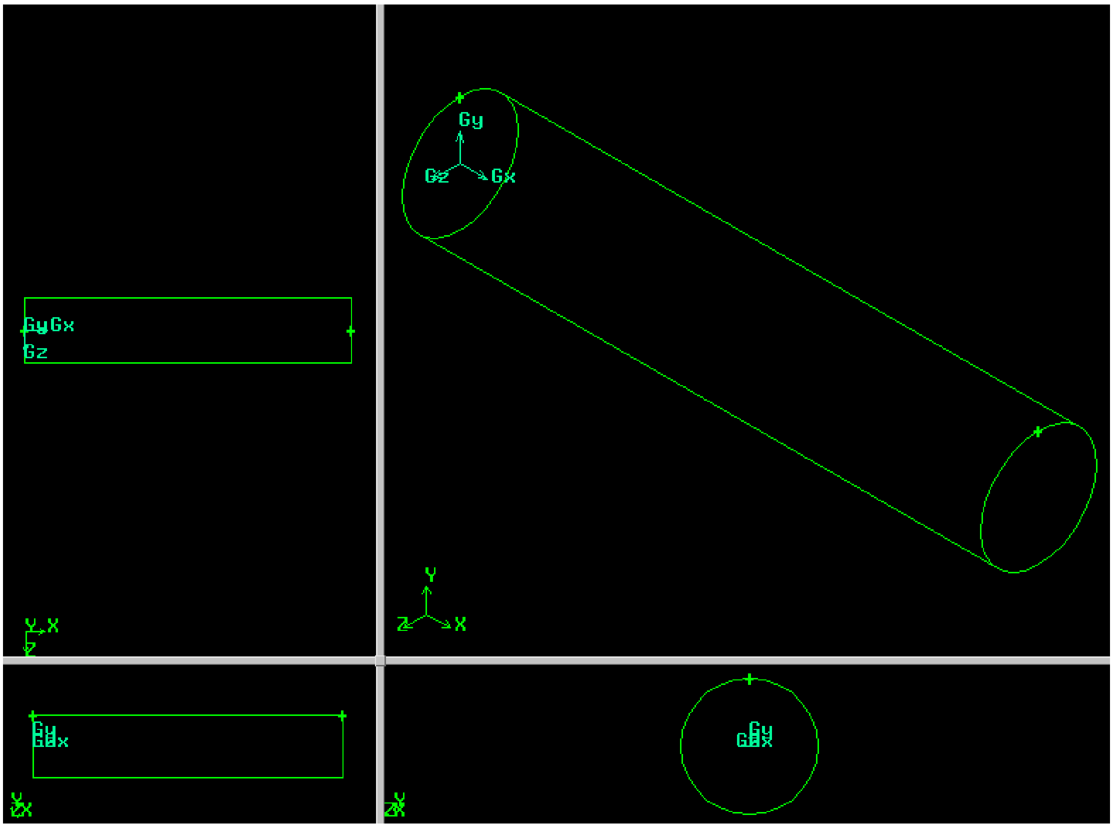


图 3.11.8

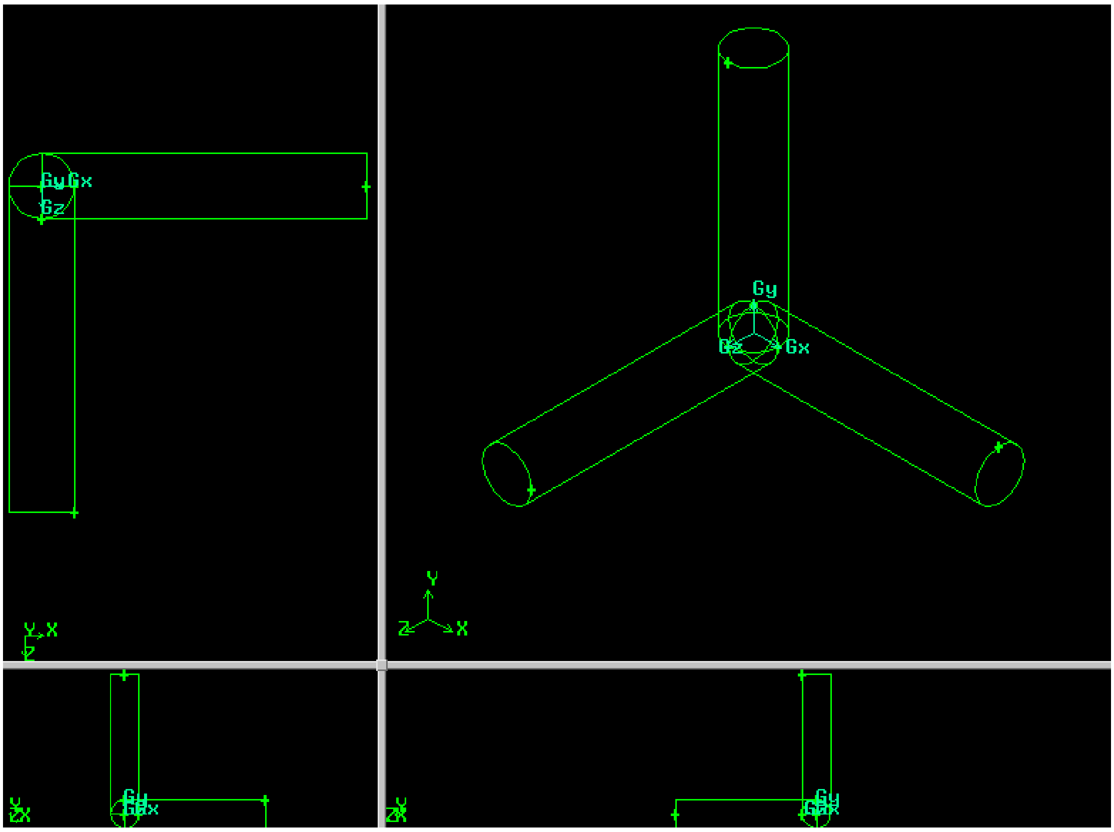


图 3.11.9

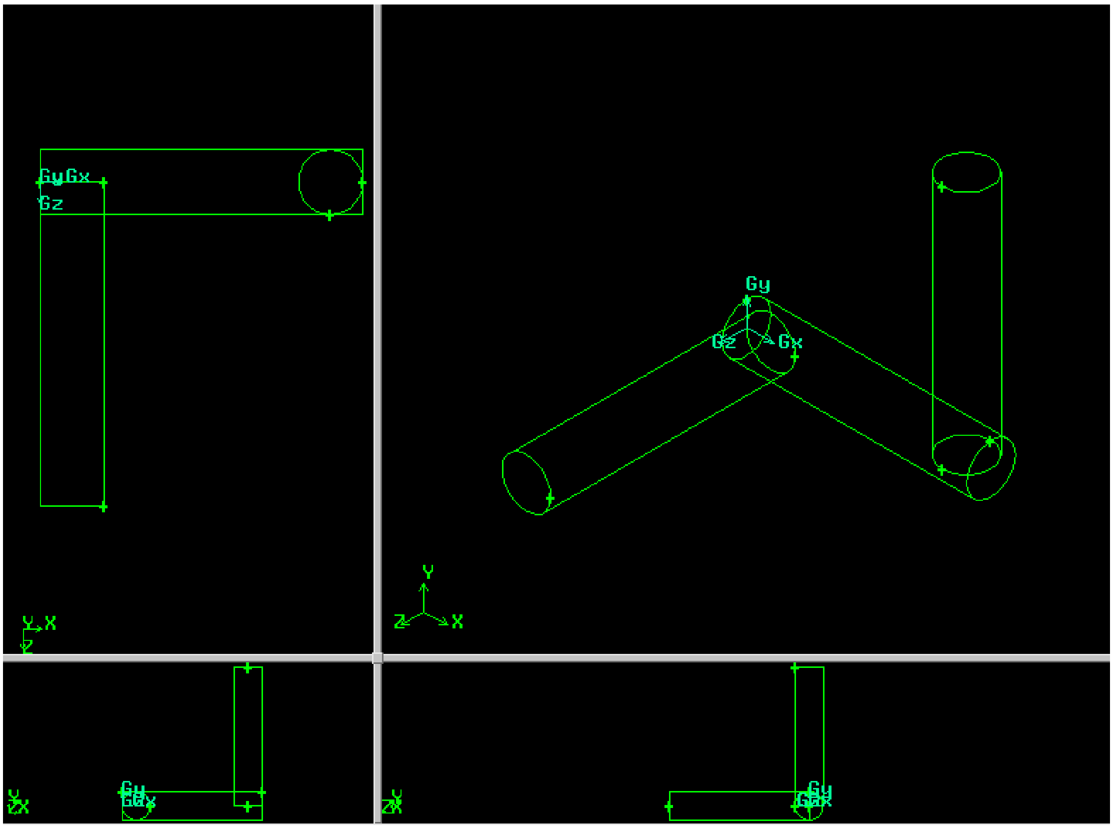


图 3.11.10

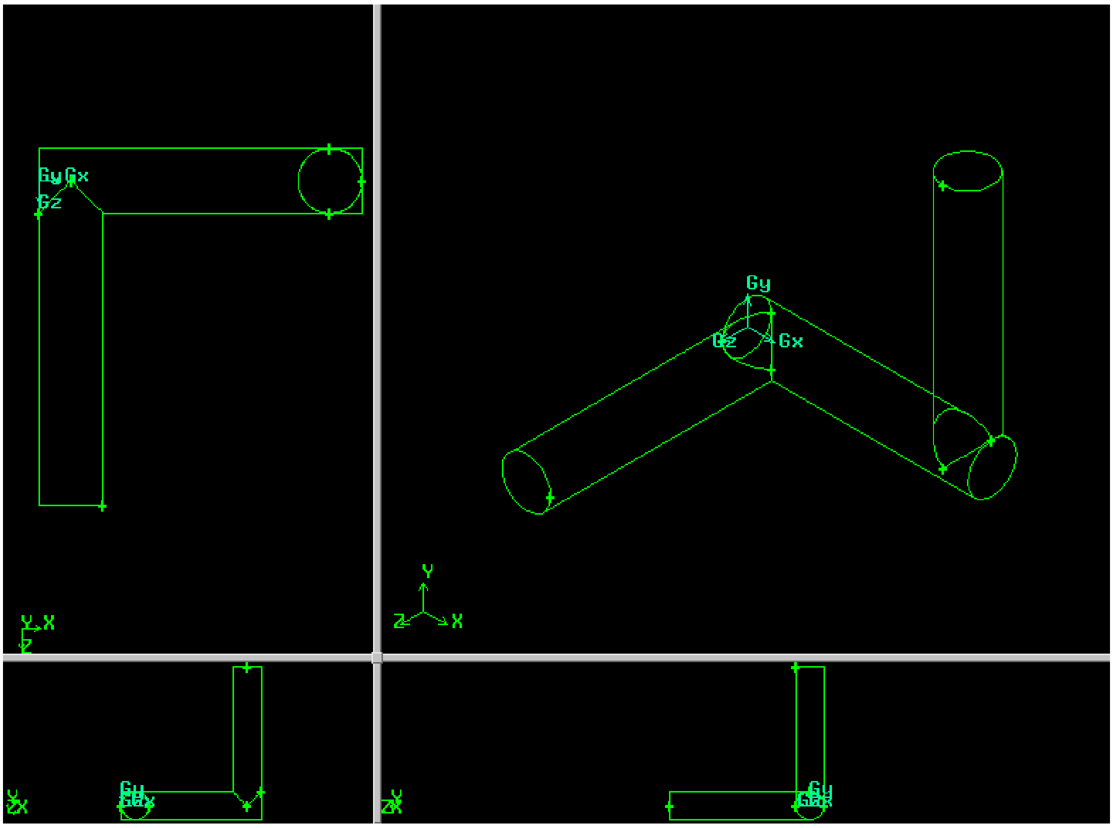
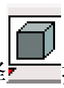


图 3.11.11

6. 更加清楚的观察三维几何体，可以选择按钮（见图 3.11.12）。

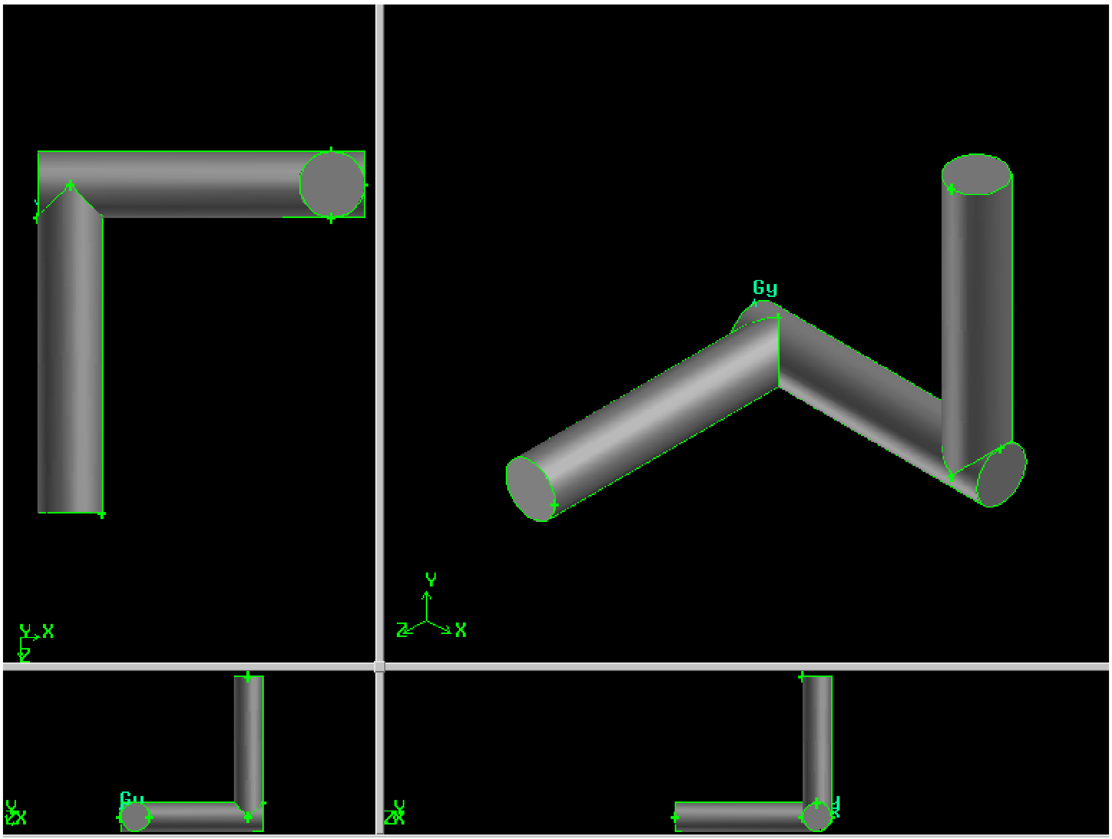
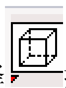


图 3.11.12

7．选择按钮即可恢复原状。

3.11.3 网格划分

三维几何体网格的划分与二维的基本一样，但三维物体的网格划分比较难以把握，尤其是对局部的加密。

3.11.4 三维双孔喷嘴

图 3.11.13 显示的最终创建的几何体。由于流场的对称性，因此取一半的流场进行计算。喷嘴上游管径为 36mm，喷嘴直径为 6mm，两喷嘴中心距为 12mm。喷嘴和上游管径连接处有 1mm 的倒角。

三维双孔喷嘴模型创建的难点还在于网格的划分。对于这种复杂几何体的组合，并不是简单的多个三维基本几何体的堆砌，而要进行布尔运算，否则在体与体的交接处就会出现两个重叠的面，导致计算时出错。而布尔运算后的几何体为一个整体，这种复杂的几何体要划分四边形网格是很困难的，这时可以再创建一些线、面，将复杂的几何体重新划分为几个标准的几何体。这些复杂的操作其目的只有一个：保证体与体的交接处只有一个面。

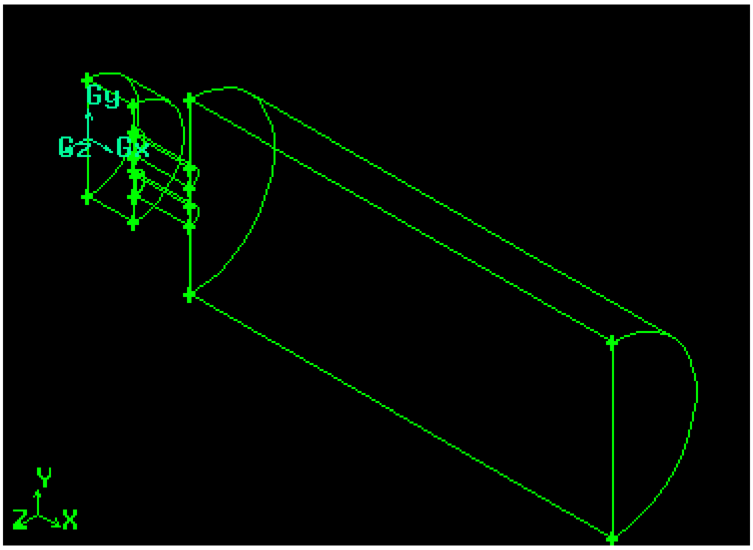


图 3.11.13

（一）创建几何体

1. 在 GAMBIT 中创建一个半径为 18，长 16 的圆柱体，圆柱体的法向指向正 X 轴。

2. 再创建两个直径为 7，6，高 1 的圆台，并将它们分别沿 Y 方向移动-6，6 个单位，沿 X 方向移动 16 个单位（见图 3.11.14）。在 GAMBIT 中创建的所有几何体的起点都在原点上。

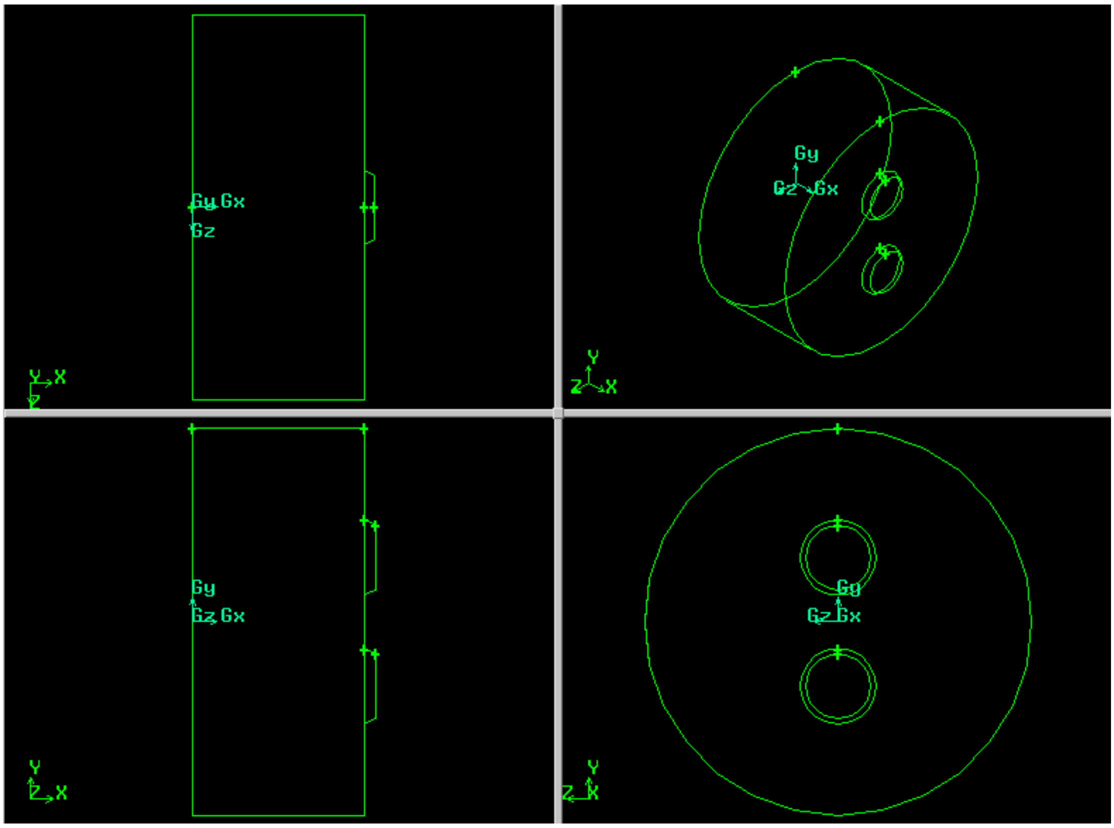
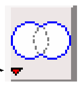


图 3.11.14

3． 利用布尔运算中的 uion 命令，将它们合成一个整体。

4． 创建两个直径 6mm，高 19mm 的圆柱，并将它们分别沿 Y 方向移动-6，6 个单位，沿 X 方向移动 17 个单位（见图 3.11.15）。

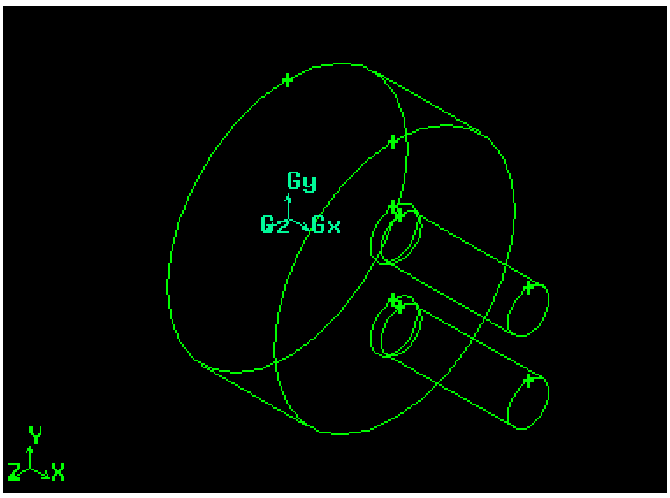


图 3.11.15

5. 创建一个直径 60mm，高 180mm 的圆柱，将其沿 X 轴移动 36mm（图 3.11.16）。

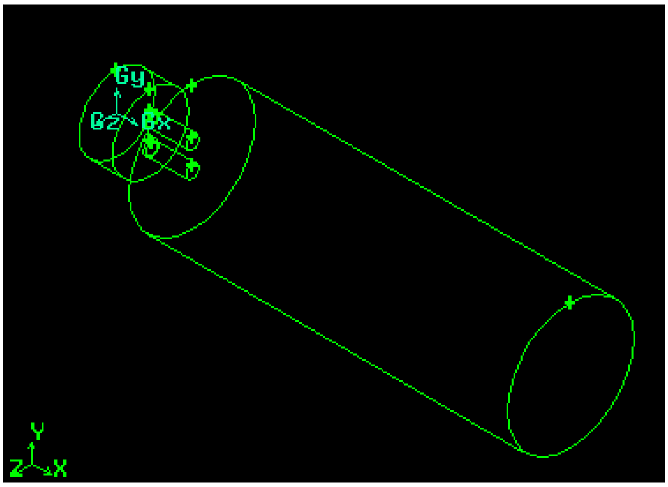


图 3.11.16

6. 利用 uion 命令，将视图中所有的几何体合成一个几何体 volume1。
7. 创建一个长 216，宽 60，高 60 的长方体，将其沿 Y 轴移动-30 个单位。
8. 利用布尔运算中的减命令，将 volume.1 削去一半（见图 3.11.17）。

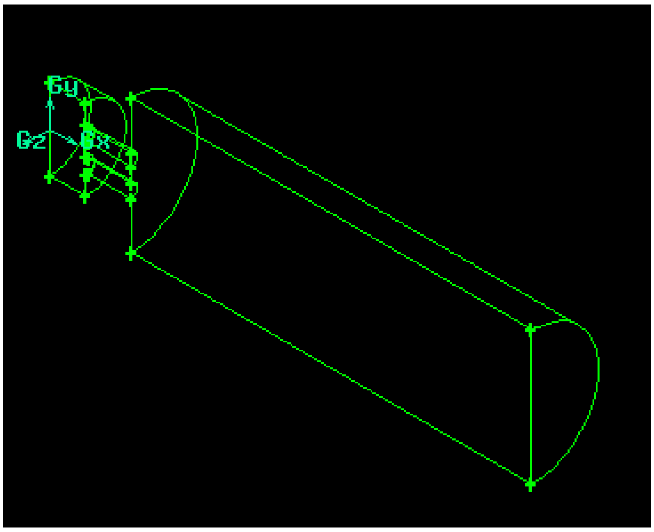


图 3.11.17

（二）重新划分几何体

由于布尔运算的结果，几何体被剖开的部分变成了一个面（红色部分）（见图 3.11.18），要将该几何体重新划分为几个标准的几何体，就必须先将这个面重新划分成几个面。

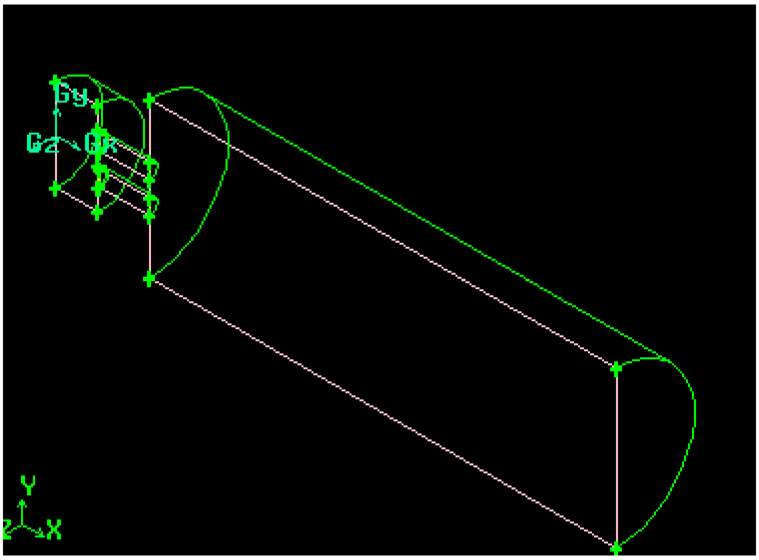


图 3.11.18

1. 如图 3.11.19 所示，创建六条新的线（黄色部分）。

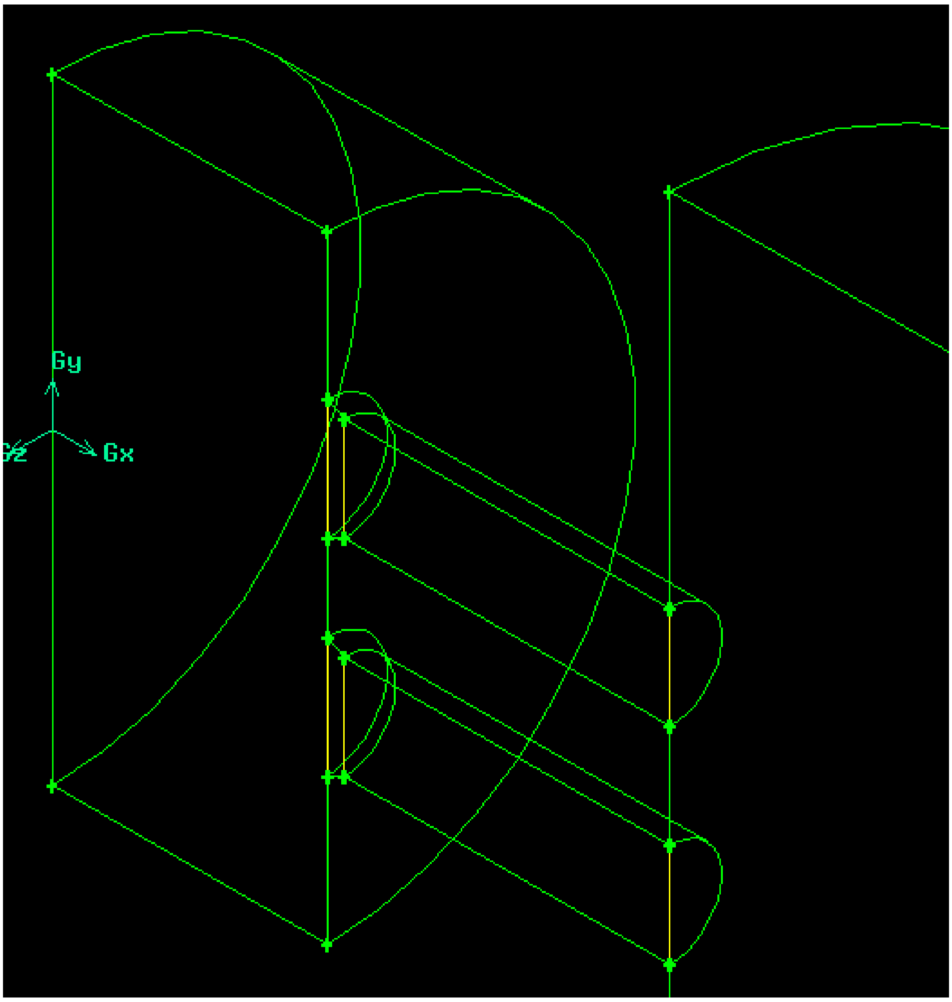


图 3.11.19

2． 将原来的对称面划分为六个面（见图 3.11.20）。



图 3.11.20

3． 将原来的几何体划分还原为六个标准的几何体（见图 3.11.21）。

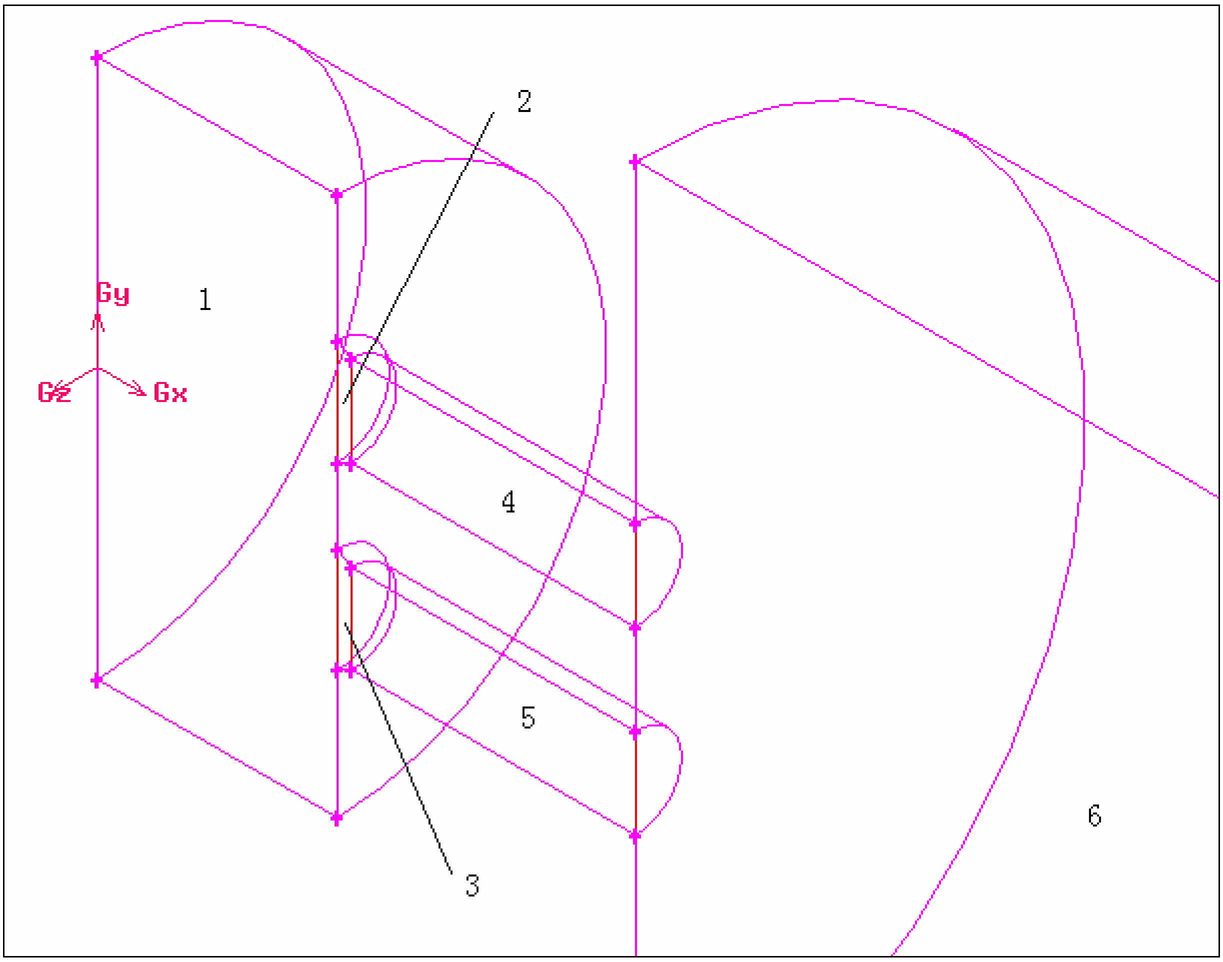


图 3.11.21

（三）划分网格

对于三维的几何体，划分四边形网格一般采用 Cooper 的方法。这就相当于三维建模中的放样，先给定首尾两个面以及路径，再创建整个几何体。具体的网格划分就不在这里赘述。

3.12　CAD/CAE 接口

引入 CAD 图形

Gambit 只适用于创建简单的三维几何体，对于复杂形体而言，其绘图功能是远远不够的，这时 Gambit 允许我们引入一些其他软件创建的文件，常用的有 Autocad 创建的 ASCI 形式的文件.sat。

CAD 中创建的图形要输出为.sat 文件，要满足一定的条件。对于二维图形来说，它必

须是一个 region，也就是说要求是一个联通域。对于三维图形而言，要求其是一个 ASCI body。

范例

一．二维轴对称维多辛斯基曲线喷嘴

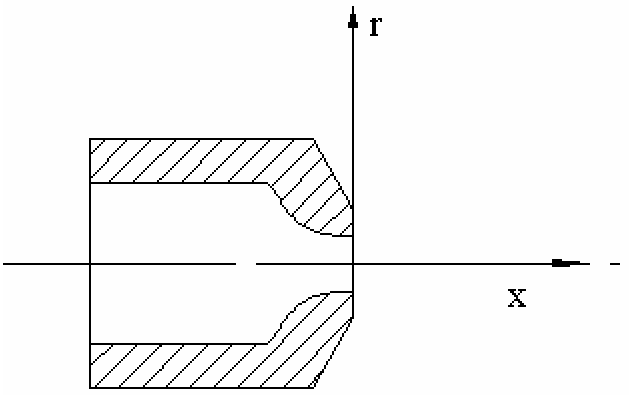


图 3.12.1 喷嘴示意图

图 3.12.1 为维多辛斯基曲线喷嘴示意图。图中的维多辛斯基曲线虽然在 gambit 中也能创建，但曲线的光滑效果不如 CAD 中的好。因此在遇到复杂几何体时，可以考虑在 CAD 中绘制部分图形然后在 GAMBIT 中进行组装。

（一） 在 Autocad 中创建维多辛斯基曲线

1． 利用 pline 命令将维多辛斯基曲线上的各点坐标连成一条折线。
2． 利用 pedit 命令使折线光滑。
3． 创建其他轮廓线（见图 3.12.2）。

喷嘴的具体参数见参考图纸。

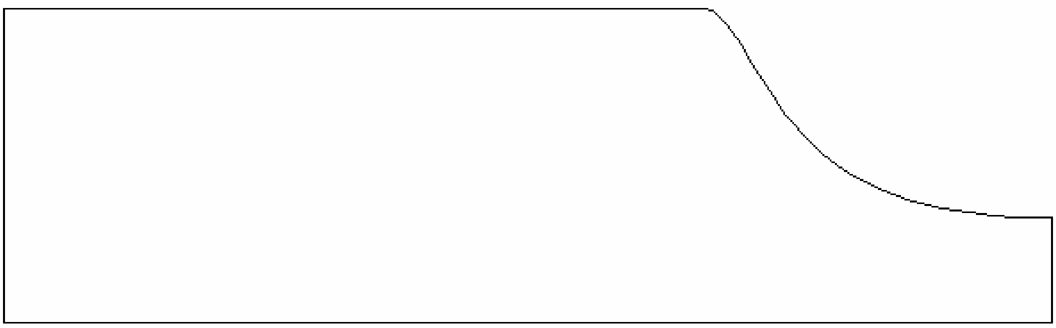
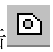


图 3.12.2 CAD 中创建的喷嘴轮廓线

（二） 输出为 ACIS 的.sat 文件

对于二维图形要输出为.sat 文件，必须是一个 region 图形。

1. 输入 region 命令，或在命令面板中单击。
2. 选择喷嘴轮廓线，单击鼠标右键或回车。
3. 选择 file/export，选择保存类型为 ACIS（*.sat），输入文件名为 jet.sat。
4. 选择喷嘴轮廓线，单击鼠标右键或回车。

（三） 在 gambit 中输入.sat 文件

1. 在 gambit 中选择 file/import/ACIS，输入文件名，单击 accept 按钮（见图 3.12.3），即可将 CAD 中创建的图形读入 gambit（见图 3.12.4）。

注意：由于 gambit 中只能利用坐标参数进行定位，所以在 CAD 中创建图形时要注意选好坐标（如起始点为原点坐标）。

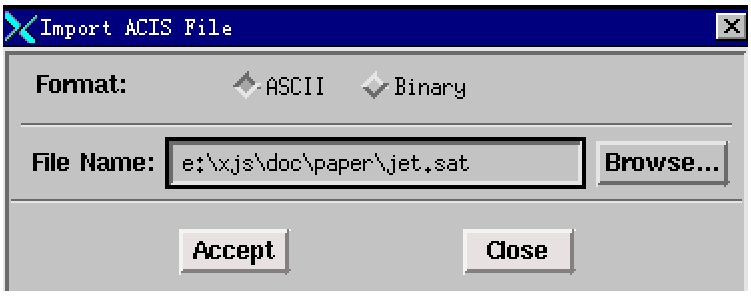


图 3.12.3 输入对话框

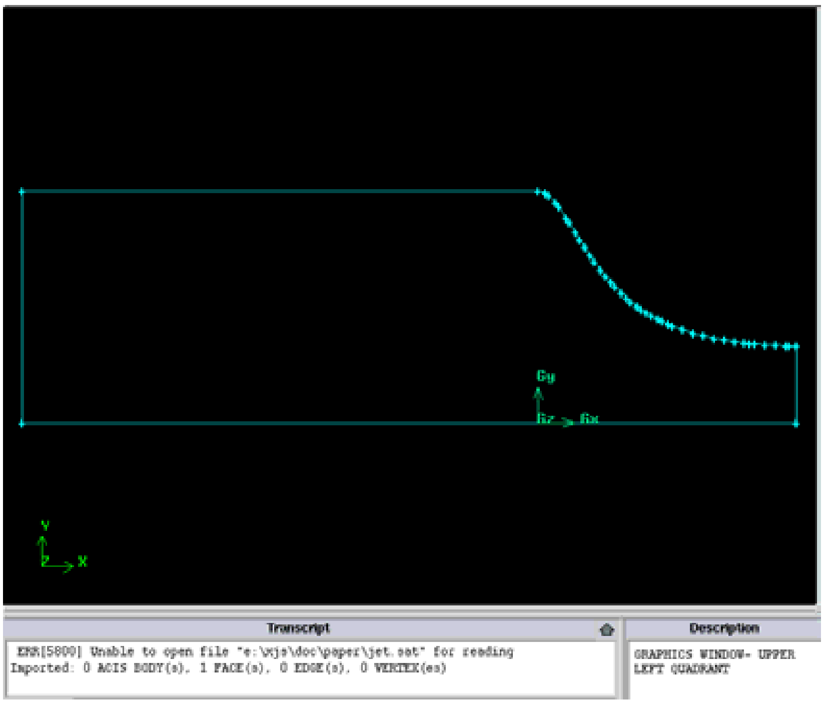


图 3.12.4 输入图形

（四） 完成模型的其他部分

1. 如图 3.12.5 所示，完成模型的其他部分，将喷嘴的外流场组成一个面。计算域为 20D*5D

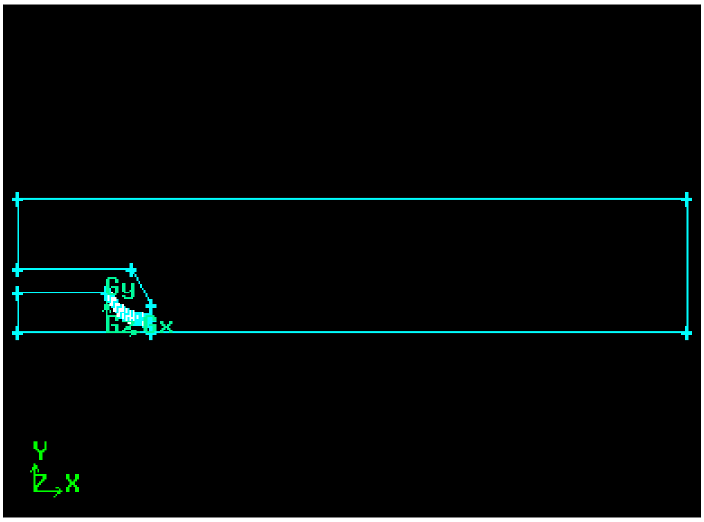


图 3.12.5 二维轴对称喷嘴计算域

1. 喷嘴内部的面（face1），定义网格数为 80*50，网格类型为四边形 map 网格（图 3.12.6）。

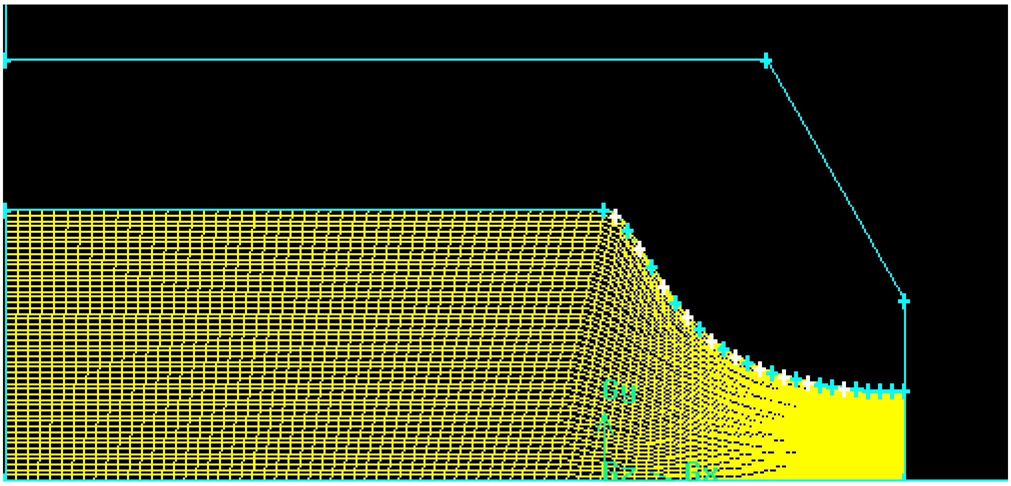


图 3.12.6

2. 喷嘴外部的面（face2），定义轴线上网格点为 240 个。定义喷嘴外轮廓线的网格点数（见图 3.12.7）

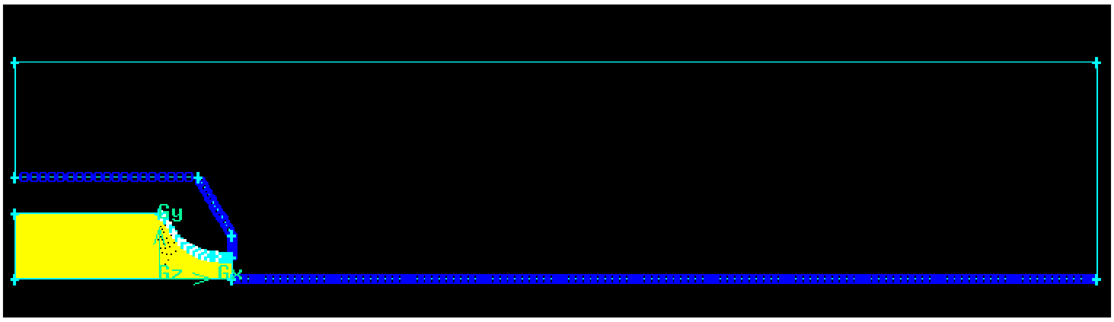


图 3.12.7

注意：对于网格的划分，如果要求控制网格的密度，可以遵循从线到面的原则，但是对于多边形区域而言，不能将所有边的网格点都定死，必须有一些边不定义网格。如四边形区域，一般只定义相邻两个边的网格。至于多边形区域怎样定义边上的网格，必须在实践中不断的尝试。

3．　划分外区域的网格，网格类型为 submap（见图 3.12.8）。

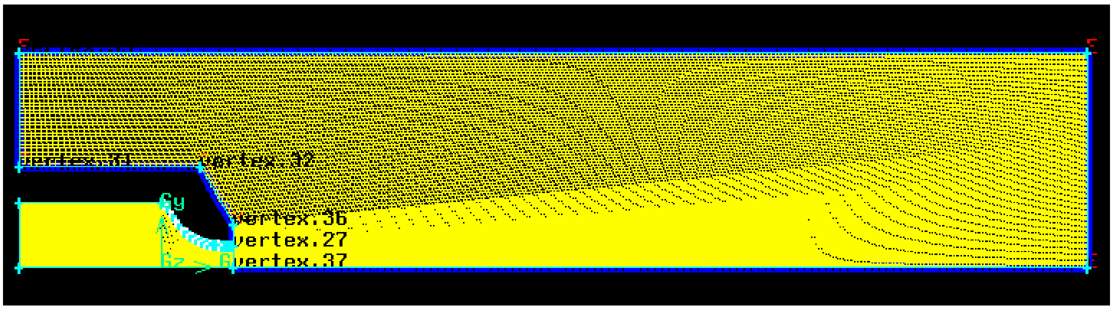
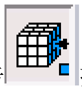


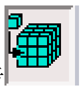
图 3.12.8

定义边界条件

1．　选择 solver/fluent5。

2．　单击按钮。

3．　定义各边界条件。

4．　单击，将两个面设为同一个连续体（fluid）（见图 3.12.9）

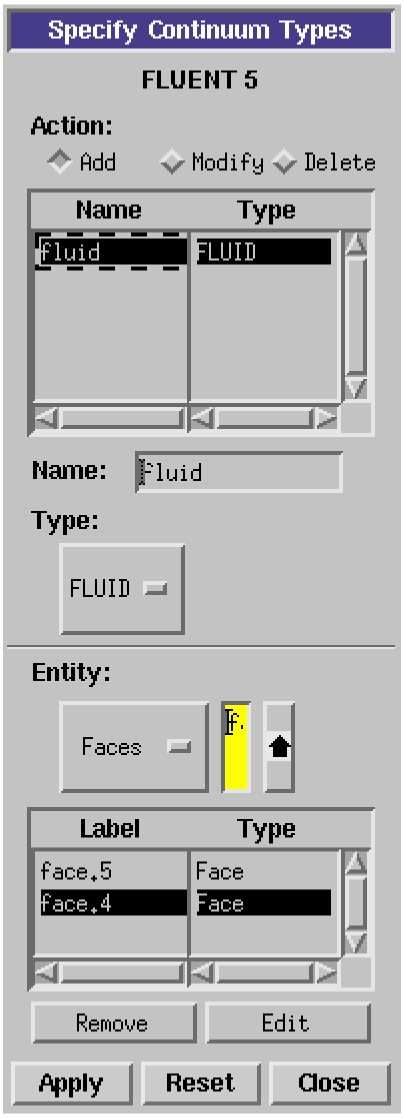


图 3.12.9

注意：对于一个复杂的几何体而言，在网格划分时必定要划分为多个区域。将这些区域定义到一个统一的连续体中，这样，不同区域间的分隔线就会被默认为内部网格点。

将网格输出为.msh 文件。

第三章　FLUENT 对网格文件的操作

Fluent 可以导入多种不同来源的各种类型的网格文件。你可以对网格进行各种各样的处理：运动或缩放结点坐标，为并行处理分离单元，重新标记区域中的网格单元以减少带宽，以及合并或分离区域。你可以获得各种各样的网格调试信息，包括内存的使用复杂度、拓扑结构和区域信息。你可以检查出网格中的结点、面和单元数目，决定区域中的最小和最大单元体，并且可以核查每个单元的合适的结点和面数。上面的以及其它的一些能力将在下面的章节中描述。

4.1 网格的拓扑结构

4.2 网格划分的要求

4.3 载入网格

4.4 非正则网格

4.5 检查网格

4.6 报告网格的统计数据

4.7 修改网格

4.8 将网格分区用于并行计算

4.1 网格的拓扑结构

Fluent 是一个基于非结构网格的求解器，它运用内部的数据结构来设计网格文件中单元、面和网格点的顺序并保持相邻单元的联系。所以它不需要用 i，j，k 标识指数来标识相邻单元的位置。由于求解器没有把一个全部结构或拓扑施加到网格上，因此使用网格拓扑有很大的弹性，这对解决问题是非常有利的。在二维（2D）中，可以使用的单元有四边形和三角形单元，在三维（3D）中，可以使用的是六面体、四面体、棱锥和楔形单元。FLUENT 允许使用的单块和多块的结构网格，四边形和三角形单元的混合网格，或六面体、四面体、棱锥和楔形单元的混合网格。另外，它还接受带有悬挂节点（即，节点：位于边或面上，但不是这些边或面所围成的所有单元的顶点）的网格，以及带有非正则边界的网格（即，网格：有多个子区域，位于内部子区域边界上的网格节点不同一）。

第 4.1.1 节给出了一些关于 FLUENT 可接受的网格例子。第 4.1.2 节介绍了怎样选择最适合你的问题的网格类型。

4.1.1 可接受的网格拓扑的例子

如上所述，FLUENT 可以解决多种网格的问题。图 4.1.2-4.1.12 给出了一些在 FLUENT 中有效使用的网格例子。O-类型的网格，壁面厚度为零的网格，C-类型的网格，正则的块结构网格，多块结构网格，非正则网格，以及非结构三角形、四面体、四边形和六面体网格都是 FLUENT 所接受的网格。

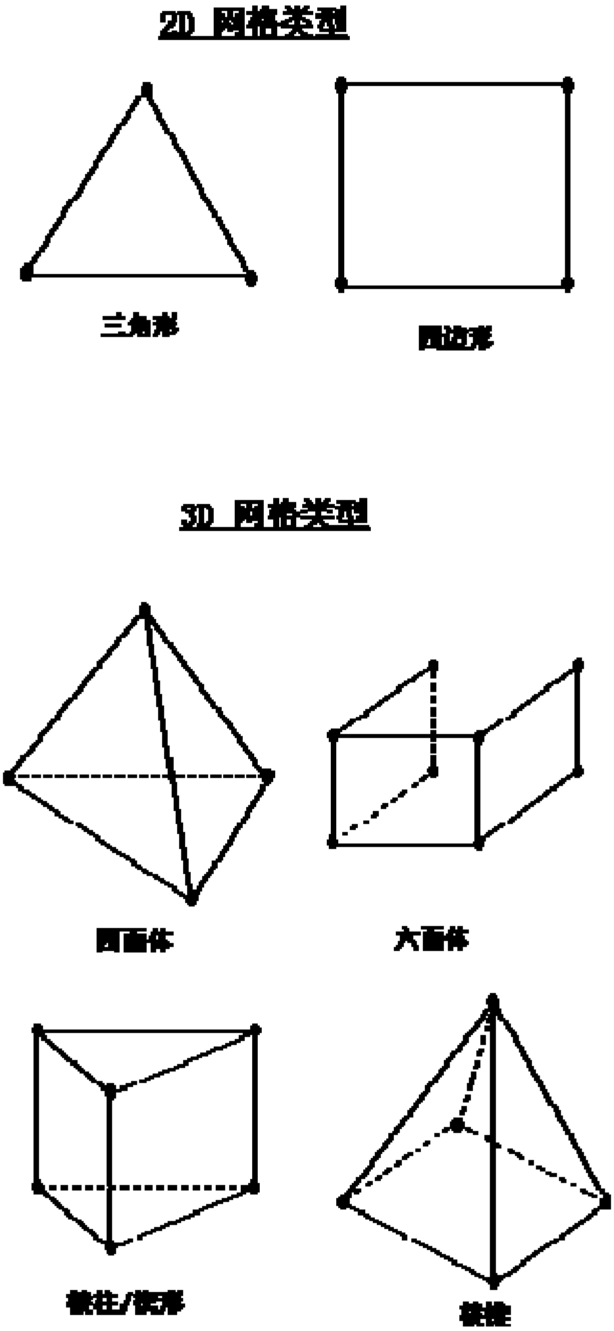


图 4.1.1：单元类型

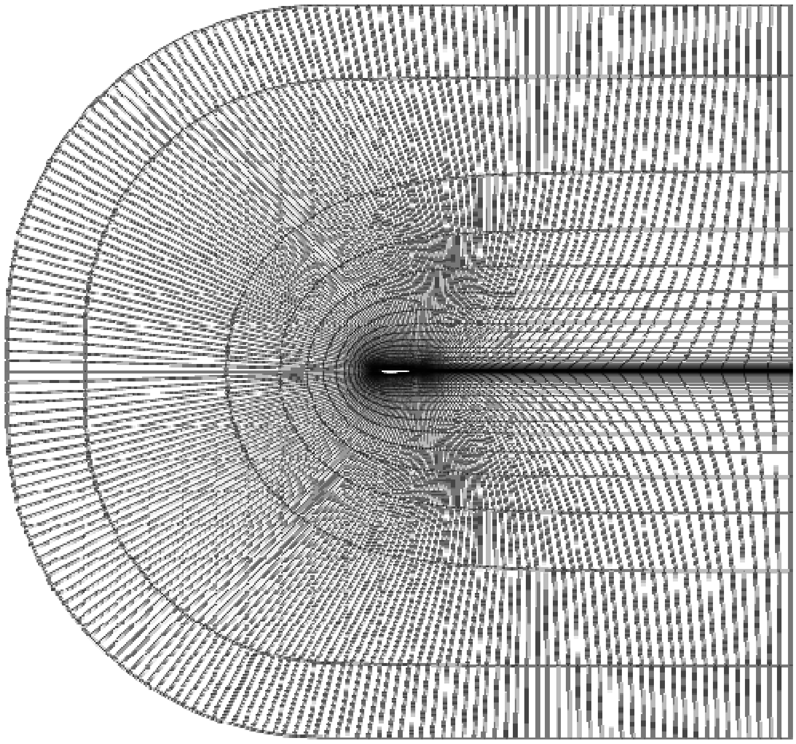


图 4.1.2：关于一个机翼的结构四边形网格

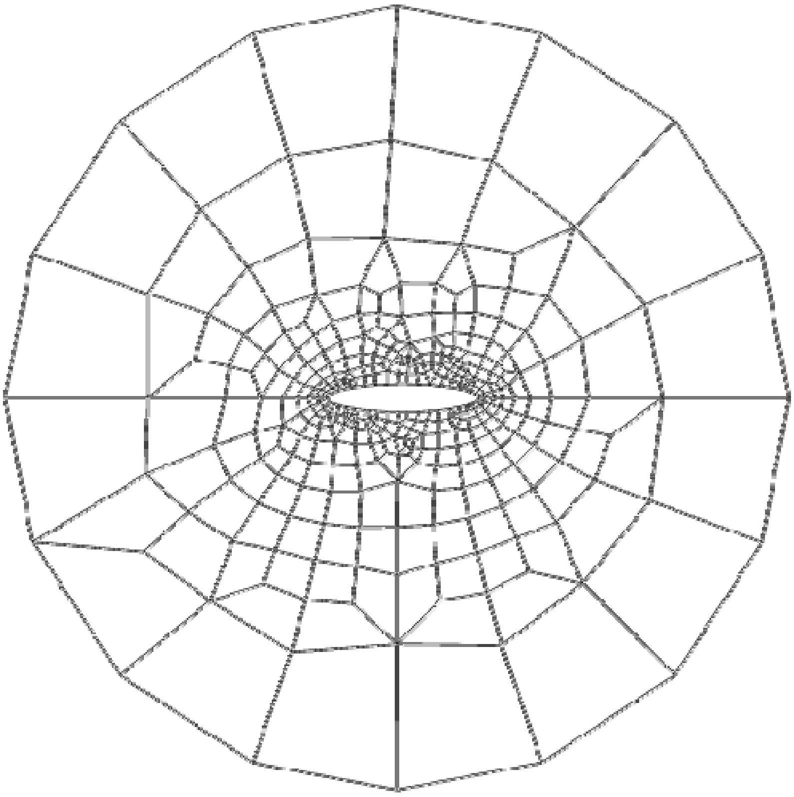


图 4.1.3：非结构四边形网格

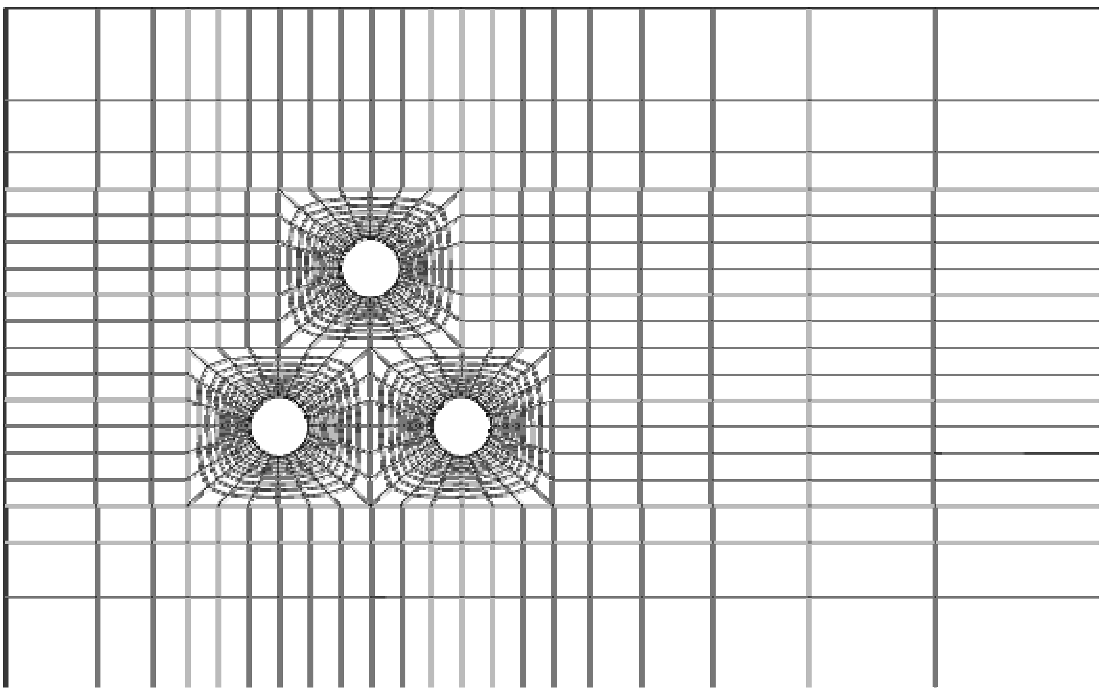


图 4.1.4：多块结构四边形网格

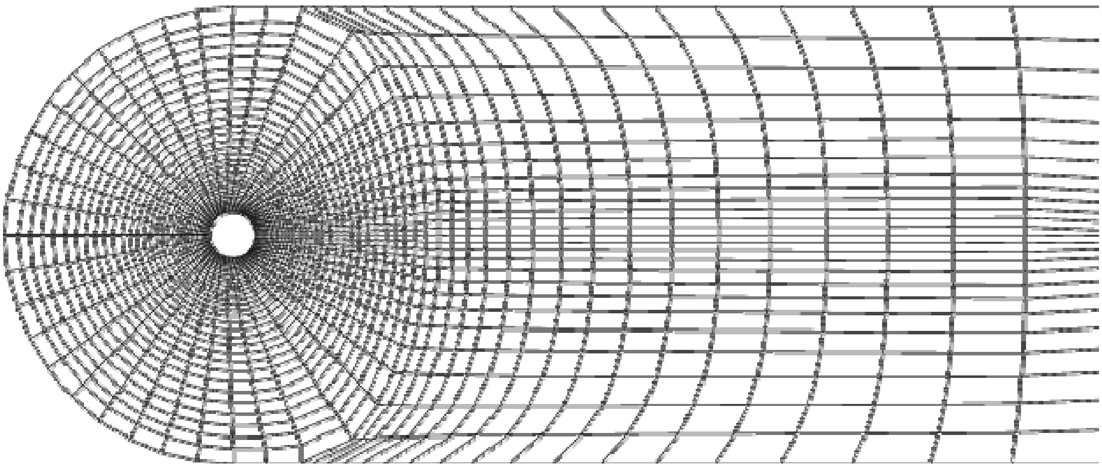


图 4.1.5：O－类型的结构四边形网格

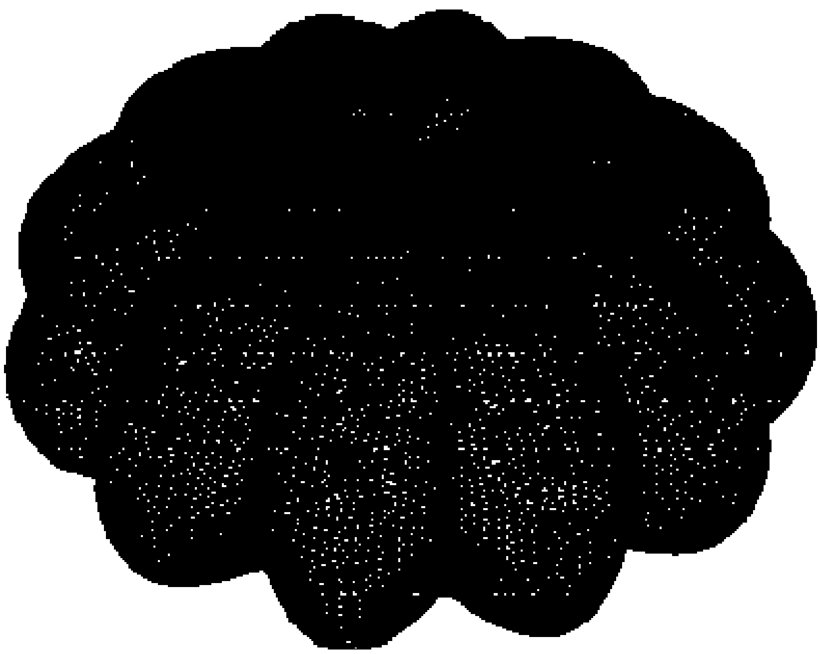


图 4.1.6：壁面厚度为零的降落伞模型

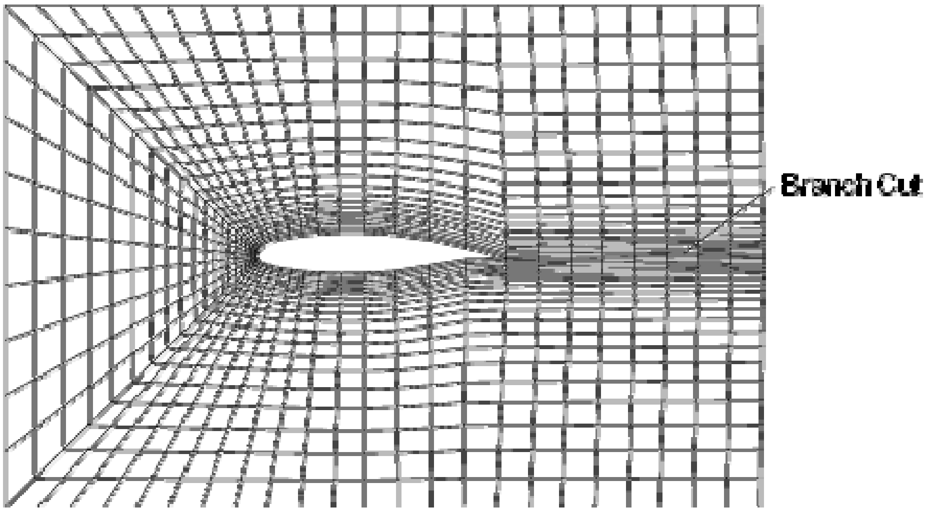


图 4.1.7：C-类型的结构四边形网格

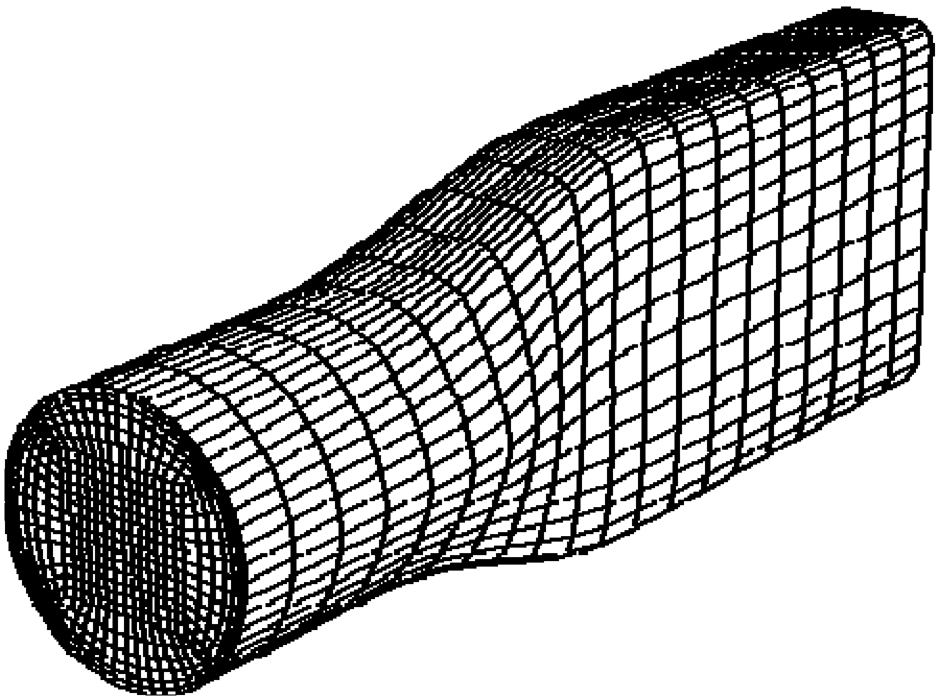


图 4.1.8：3D 多块结构网格

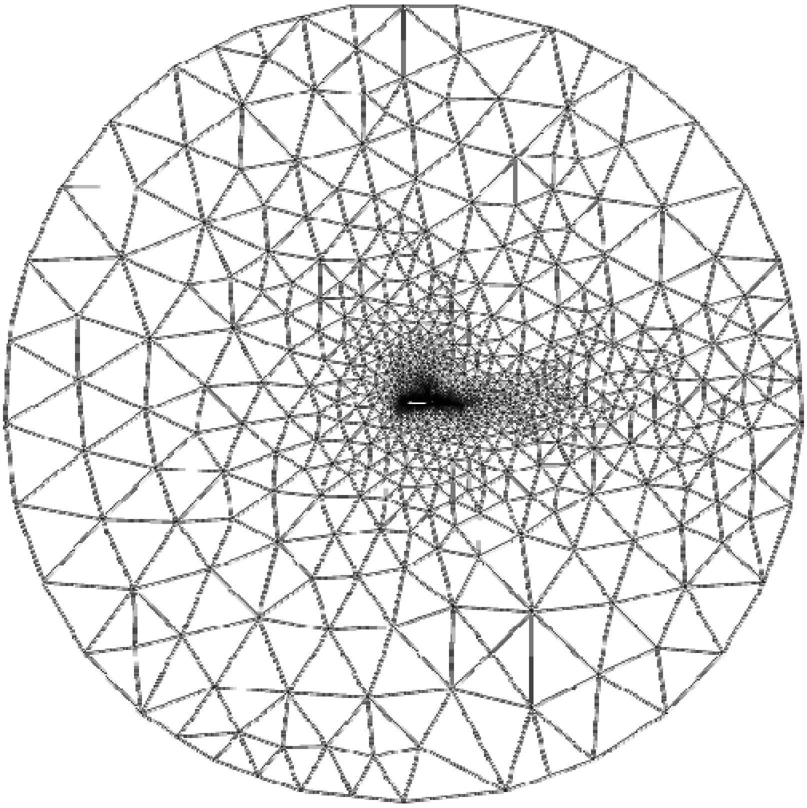


图 4.1.9：关于一个机翼的非结构三角形网格

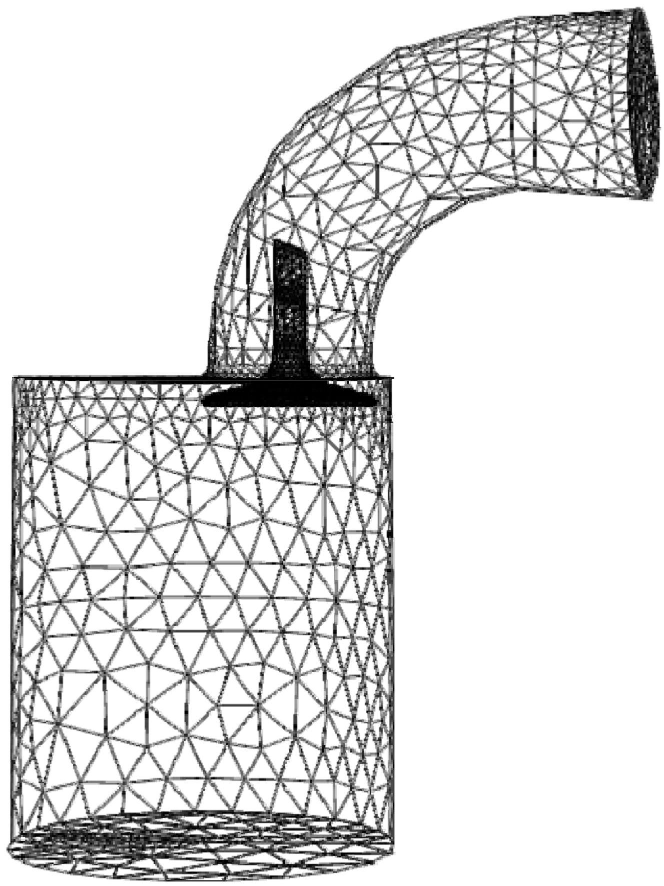


图 4.1.10：非结构四面体网格

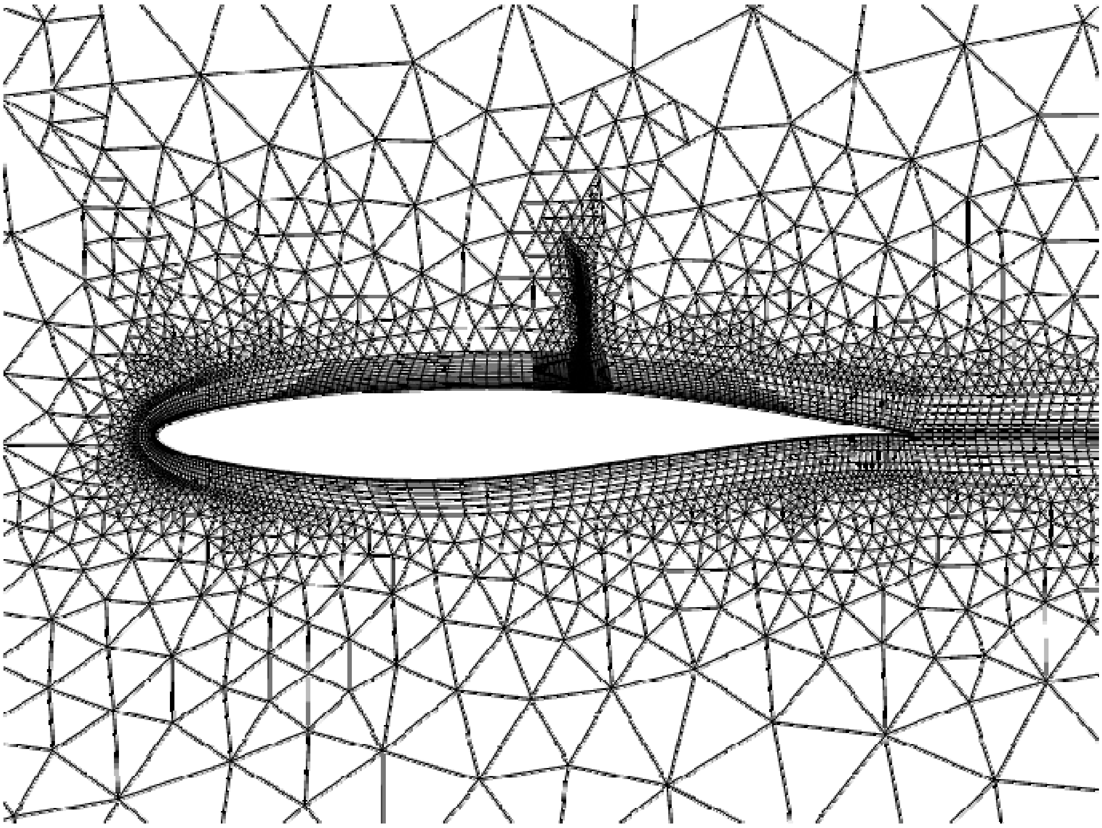


图 4.1.11：带有悬挂节点的三角形与四边形的混合网格

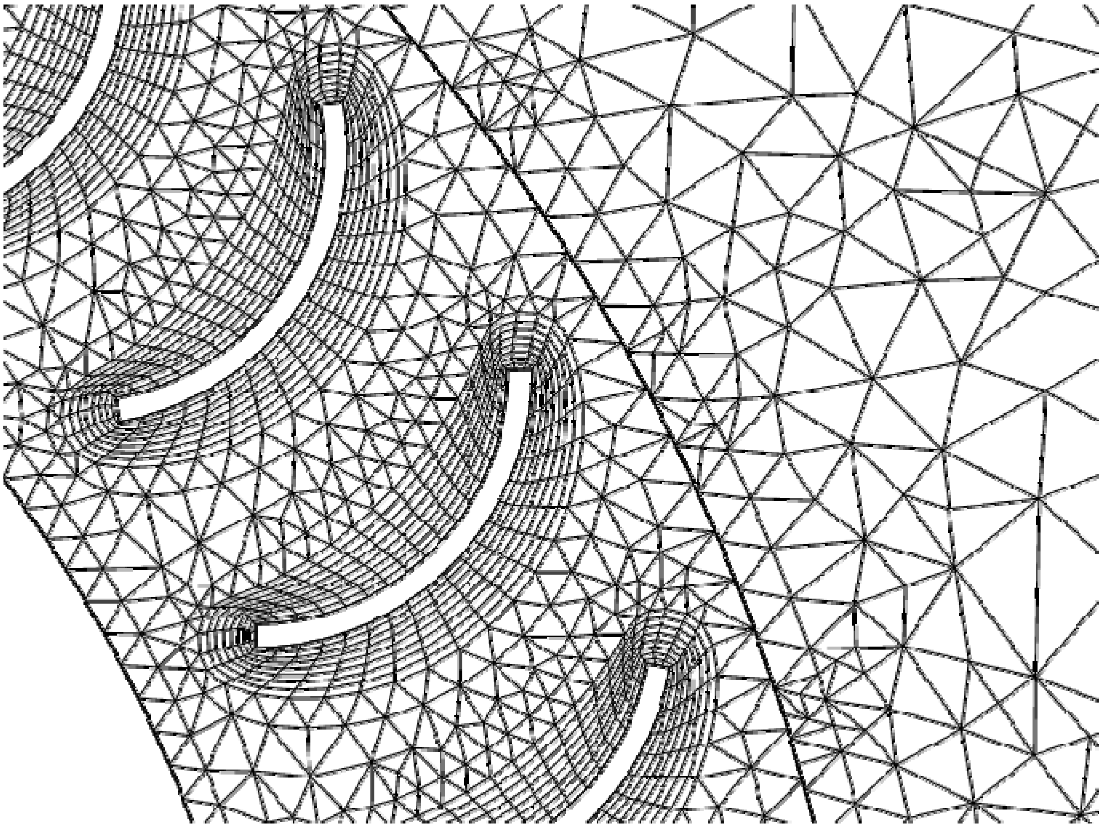


图 4.1.12：关于一个固定片回转轴的几何的非正则混合网格

4.1.2　选择适当的网格类型

在 2D 中，FLUENT 可以使用三角形和四边形单元以及它们的混合单元所构成的网格。在 3D 中，它可以使用四面体，六面体，棱锥，和楔形单元所构成的网格。选择那种类型的单元取决于你的应用。当选择网格类型的时候，应当考虑以下问题：

设置的时间（setup time）

计算的费用（computational expense）

数值的耗散（numerical diffusion）

设置的时间

在工程实践中，许多流动问题都涉及到比较复杂的几何形状。一般来说，对于这样的问题，建立结构或多块（是由四边形或六面体元素组成的）网格是极其耗费时间的。所以对于复杂几何形状的问题，设置网格的时间是使用三角形或四面体单元的非结构网格的主要动机。然而，如果所使用的几何相对比较简单，那么使用哪种网格在设置时间方面可能不会有明显的节省。

如果你已经有了一个建立好的结构代码的网格，例如 FLUENT 4，很明显，在 FLUENT 中使用这个网格比重新再生成一个网格要节省时间。这也许是你在 FLUENT 模拟中使用四边形或六面体单元的一个非常强的动机。注意，对于从其它代码导入结构网格，包括 FLUENT 4，FLUENT 有一个筛选的范围。

计算的费用

当几何比较复杂或流程的长度刻度的范围比较大的时候，可以创建是一个三角形/四面体网格，因为它与由四边形/六面体元素所组成的且与之等价的网格比较起来，单元要少的多。这是因为一个三角形/四面体网格允许单元群集在被选择的流动区域中，而结构四边形/六面体网格一般会把单元强加到所不需要的区域中。对于中等复杂几何，非结构四边形/六面体网格能构提供许多三角形/四面体网格所能提供的优越条件。

在一些情形下使用四边形/六面体元素是比较经济的，四边形/六面体元素的一个特点是它们允许一个比三角形/四面体单元大的多的纵横比。一个三角形/四面体单元中的一个大的纵横比总是会影响单元的偏斜（skewness），而这不是所希望的，因为它可能妨碍计算的精确与收敛。所以，如果你有一个相对简单的几何，在这个几何中流动与几何形状吻合的很好，例如一个瘦长管道，你可以运用一个高纵横比的四边形/六面体单元的网格。这个网格拥有的单元可能比三角形/四面体少的多。

数值的耗散

在多维情形中，一个错误的主要来源是数值耗散，术语也为伪耗散(false diffusion)。之所以称为“伪耗散”是因为耗散不是一个真实现象，而是它对一个流动计算的影响近似于增加真实耗散系数的影响。

关于数值耗散的观点有：

当真实耗散小，即情形出现对流受控时，数值的耗散是最值得注意的。

关于流体流动的所有实际的数值设计包括有限数量的数值耗散。这是因为数值耗散起于切断错误，而切断错误是一个表达离散形式的流体流动方程的结果。

用于 FLUENT 中的二阶离散方案有助于减小数值耗散对解的影响。

数值耗散的总数反过来与网格的分解有关。因此，处理数值耗散的一个方法是改进网格。

当流动与网格相吻一致时，数值耗散减到最小。

最后这一点与网格的选择非常有关。很明显，如果你选择一个三角形/四面体网格，那么流动与网格总不能一致。另一方面，如果你使用一个四边形/六面体网格，这种情况也可能会发生，但对于复杂的流动则不会。在一个简单流动中，例如过一长管道的流动，你可以依靠一个四边形/六面体网格以尽可能的降低数值的耗散。在这种情形，使用一个四边形/六面体网格可能有些有利条件，因为与使用一个三角形/四面体单元比起来，你将能够使用比较少的单元而得到一个更好的解。

网格划分的要求

这节介绍了特殊的几何/网格要求和关于网格质量的总的注解。

4.2.1 几何/网格的要求

在你的问题建立的开始，你应当注意对几何建立和网格结构的以下几点要求：

轴对称几何必须如此定义：在 Cartesian 坐标系下，旋转轴是 x 轴（如图 4.2.1）。

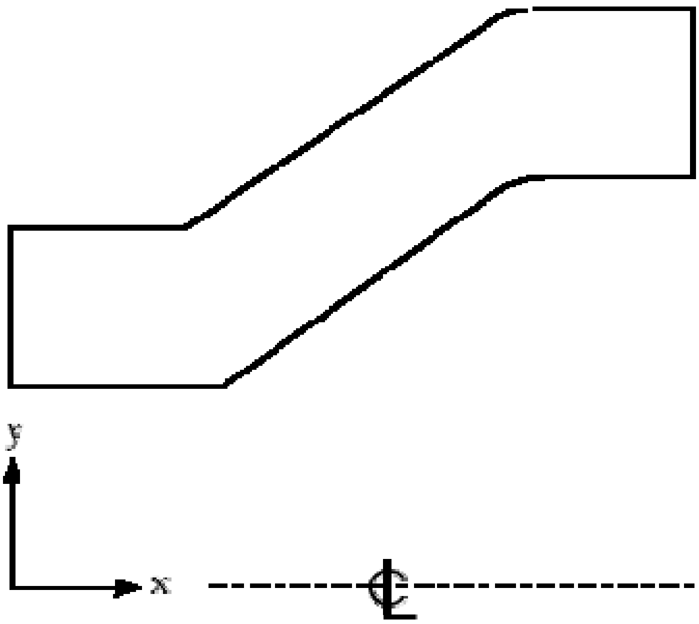


图 4.2.1：以 x 轴作为中心线的轴对称几何的建立

FLUENT 允许使用正则或非正则的周期域来建立周期边界。对于正则的周期边界，周期域必须有同一的网格。当你生成体网格的时候，正则的周期边界可以在 GAMBIT 或 TGrid 里建立。二者选一，你可以使用 make-periodie text 命令在 FLUENT 中建立正则的周期边界。

虽然 GAMBIT 和 TGrid 能够生成正确的周期边界，但多数 CAD 包则不能够。如果你网格在这样的包中被建立，那么你可以在 FLUENT 中使用非正则的周期选项建立周期边界。然而，此选项仅建议用于平面的周期域。

4.2.2 网格质量

网格的质量对于数值计算的正确域稳定起着非常重要的作用。网格质量的特征是节点

的分布，光滑，和偏斜。

节点的密度和聚类

由于你是分离地定义一个连续区域，所解决的流动的突出特性之程度（例如，剪切层，分离的阿区域，边界层和混合域）取决于节点在网格中的密度和分布。在许多情况下，临界区域中的不良分解能够明显地改变流动的特性。

边界层（即，近壁面的网格间隔）的分解在计算壁面剪切应力和热传递系数的准确性方面也起到了很重要的作用。尤其对网格邻近壁面符合表达式（4.2.1）的层流，它更说明了这一点。

$$y_p \sqrt{\frac{u_\infty}{\nu x}} \leq 1 \qquad (4.2.1)$$

这里，y_p 是从邻近单元的中心到壁面的距离，u_∞ 是自由流速度，ν 是流体的运动粘性系数，x 是从边界层的始点沿壁面的距离。

对于湍流流动，网格的适当解也是非常重要的。由于平均流动与湍流的强烈的交互作用，湍流流动的数值结果比层流流动更易受网格从属物的影响。在近壁区，使用不同的近壁模型，需要不同的网格分解。

一般说来，流动通道不应少于 5 个单元。多数情况下则要求足够多的单元来解决通道。在大梯度的区域，像剪切层或混合域中，网格应密到足以使得在单元到单元的流动变量中的变化减到最小。不幸的是，预先决定重要流动特性的位置往往是非常困难的。进一步讲，在多数复杂的三维流场中的网格分解将受 CPU 的时间和计算机资源的局限性（即，内存和磁盘空间）的限制。虽然随着网格的增大，正确性提高，但是解的计算和结果的后处理对 CPU 和内存的要求也相应的提高。基于进化流场，解适应网格能够被用来提高或降低网格的密度，从而提供了比较经济的使用网格点的潜力（减少时间和资源要求）。

光滑（smoothness）

相邻单元间的单元体中的快速变化转变成比较大的切断错误。切断错误在控制方程及其离散近似值中的偏导数之间是不同的。FLUENT 基于单元体或单元体的梯度中的变化来优化单元，从而提供了改善网格光滑性的能力。

单元形状

单元的形状对数值解的正确性也有重要的影响。偏斜可以定义为单元的形状和与之等价的体积的一个等边单元的形状之间的偏差。歪斜厉害的单元能够降低解的正确性和稳定性。例如，理想的四边形网格将有接近 90 度的顶角，而三角形网格比较适宜有接近 60 的角和所有的角都应小于 90 度。

纵横比是单元伸展的一个度量。

流场的从属物

分解，光滑和单元的形状对解的正确与稳定的影响在很大程度上取决于所模拟的流场。例如，歪斜厉害的单元在良好的流动区域内是可以承受的，但在流动梯度大的区域中则破坏性比较严重。由于流动梯度大的位置不能预先被决定，因此你应当努力在整个流动区域中获得一个高质量的网格。

4.3　载入网格

由于 FLUENT 能够处理大量不同的网格拓扑，因此 FLUENT 中有许多源，使你从中可以得到一个用于你的模拟中的网格。你可以使用 GAMBIT，TGrid，GeoMesh，preBFC，ICEMCFD，I-DEAS，NASTRAN，PATRAN，ARIES，ANSYS，或其它预处理程序来生成一个网格，或者使用含在 FLUENT/UNS，RAMPANT，或 FLUENT 4 case 文件中的网格。你也可以准备若干个网格文件并把它们合并起来建立一个单独的网格。

4.3.1　GAMBIT 网格文件

你可以使用 GAMBIT 建立 2D 和 3D 的结构/非结构/混合网格。你可以按照 GAMBIT Modeling Guide（GAMBIT 建模指南）里所叙述的过程来建立上述的任一种网格，在 FLUENT 5/6 的格式下导出你的网格。所有这些网格可以使用 File/Read/Case…菜单项直接载入 FLUENT。

4.3.2　GeoMesh 网格文件

你可以使用 GeoMesh 建立完整的 2D 四边形或三角形网格，3D 六面体网格，和关于 3D 四面体网格的三角形表面网格。你可以按照 GeoMesh User's Guide（GeoMesh 用户指南）里所叙述的过程来建立上述的任一种网格。为完善一个 3D 四面体网格的生成，你必须把表面网格读进 TGrid，然后在那里生成体网格。然而，所有其它网格可以使用 File/Read/Case…菜单项直接载入 FLUENT。

4.3.3　TGrid 网格文件

你可以使用 TGrid 从边界或表面网格建立 2D 和 3D 非结构三角形/四面体网格。你可以按照 TGrid User's Guide（TGrid 用户指南）里所叙述的过程来建立上述的任一种网格，使用 File/Read/Case…菜单项保存你的网格。使用 File/Read/Case…菜单项直接把网格载入

FLUENT。

4.3.4 preBFC 网格文件

你可以使用 preBFC 为 FLUENT 建立两种不同类型的网格：结构四边形/六面体和非结构三角形/四面体网格。生成和载入每一网格的过程如下所述。

结构网格文件

要生成一个 2D 或 3D 结构网格，其具体过程请参阅 preBFC User’s Guide（preBFC 用户指南）。生成的网格将包含四边形（2D）或六面体（3D）元素。记得规定的壁面域和进口域分别不要超过 70 和 35。

使用 File/Import/preBFC Structured Mesh…菜单项载入网格。

要手工地把 preBFC 格式里的一个文件转换成一个适合 FLUENT 的 mesh 文件，请输入命令：

utility fl42seg input-filename output-filename

可以使用 File/Read/Case…菜单项把产生的输出文件读进 FLUENT。

非结构三角形和四面体网格文件

要生成一个 2D 非结构网格，其具体过程请参阅 preBFC User’Guide（preBFC 用户指南）。使用 MESH-RAMPANT/TGRID 命令以 RAMPANT 格式保存你的网格文件。注意，当前的 FLUENT 格式与 RAMPANT 格式是一样的。所生成的网格将包含三角形元素。使用 File/Read/Case…菜单项载入网格。

要生成一个 3D 非结构四面体网格，其具体过程请参阅关于生成一个表面网格的 preBFC User’Guide（preBFC 用户指南）。然后你将把表面网格读进 TGrid，并在那里完善网格的生成。

4.3.5 ICEMCFD 网格文件

你可以使用 ICEMCFD 建立 FLUENT 4 格式的结构网格和 RAMPANT 格式的非结构网格。要载入一个 FLUENT 4 的网格，请按照第 4.3.8 节中的指导进行。使用 File/Read/Case…菜单项载入一个 RAMPANT 网格。注意，当前的 FLUENT 格式与 RAMPANT 格式是一样的，而与 FLUENT 4 的格式不一样。在读进一个三角形和四面体 ICEMCFD 体单元后，你应当执行平滑（smoothing）和交换（swapping）操作以改进网格的质量。

4.3.6 来自第三方的 CAD 包的网格文件

FLUENT 能够从一些第三方的 CAD 包载入网格文件，包括 I-DEAS，NASTRAN，PATRAN，和 ANSYS。

I-DEAS Universal 文件

你可以使用三种不同的方法把 I-DEAS 通用文件载入到 FLUENT 中：

你可以生成一个包含三角形，四边形，四面体，楔形或六面体元素的 I-DEAS 表面和体网格，并使用 TGrid User’s Guide（TGrid 用户指南）中所述的命令把网格载入到 TGrid 中。如果有必要，还要完善网格的生成。

你可以生成一个线性的三角形，四边形，四面体，楔形，或六面体元素的 I-DEAS 体单元，并使用 File/Import/IDEAS Universal…菜单项直接载入网格。

你可以生成一个线性的三角形，四边形，四面体，楔形，或六面体元素的 I-EAS 体网格，然后使用过滤器 fe2ram 把 Universal 文件转换成 FLUENT 使用的格式。关于把一个 I-DAS Universal 格式的输入文件转换成一个 FLUENT 格式输出文件的内容，在本节的后面有介绍。写完输出的文件之后，你可以使用 File/Read/Case…菜单项把文件读进 FLUENT。

（1）识别 I-DEAS 数据集

下面的 Universal 文件数据集是通过 FLUENT 网格载入工具识别的。

Node Coordinates 数据集的数是 15，81，2411

Elements 数据集的数是 71，780，2412

Permanent Groups 数据集的数是 752，2417，2429，2430，2432，2435

对于 2D 体网格，元素必须在一个恒定的 z 平面中。注意，网格面积/网格体积的数据集是不被识别的。这意味着把多网格面积/网格体积写入一个单独的 Universal 文件中可能会给 FLUENT 带来麻烦。

（2）把节点分组以建立面域

使用 **Group** 命令在 I-DEAS 中对节点分组以建立边界面域。在 FLUENT 中，边界条件适用于每一个域。包含一个组中节点的面被集合到一个单独的域中。如果一个内部的面上有边界面的节点，那么这个内部面的节点进行分组是很重要的。

在曲线或网格区域的基础上，机械地生成组，即每一个曲线或网格区域将成为 FLUENT 中的一个不同的域，这是分组的一个方法。另外，你也可以手工地建立组，生成由与一个给定的曲线（2D）或网格区域（3D）有关的所有节点构成的组。

（3）删除副节点

I-DEAS 可能在创建元素的过程中生成副的或重合节点。在写载入到 FLUENT 中的通用文件之前，这些节点必须从 I-DEAS 中删除。

NASTRAN 文件

你可以使用三种不同的方法把一个 NASTRAN 文件载入到 FLUENT 中。

你可以生成一个包含三角形，四边形，四面体，楔形或六面体元素的 NASTRAN 表面或体网格，并使用 TGrid User’s Guide（TGrid 用户指南）中所述的命令把网格载入到 TGrid 中。如果有必要，还要完善网格的生成。

你可以生成一个线性的三角形，四边形，四面体，楔形，或六面体元素的 NASTRAN 体单元，并使用 File/Import/NASTRAN…菜单项直接载入网格。

你可以生成一个线性的三角形，四边形，四面体，楔形，或六面体元素的 I-EAS 体网格，然后使用过滤器 fe2ram 把 NASTRAN 文件转换成 FLUENT 使用的格式。关于把一个 NASTRAN 格式的输入文件转换成一个 FLUENT 格式的输出文件的内容，在本节的后面有介绍。写完输出的文件之后，你可以使用 File/Read/Case…菜单项把文件读进 FLUENT。

在使用上面所述的方法 2 和 3 读进一个三角形和四面体的 NASTRAN 体网格之后，你应当执行平滑和交换命令以改进网格的质量。

（1）识别 NASTRAN 数据集

下面的 NASTRAN 文件数据集是通过 FLUENT 网格载入工具识别的。

GRID 单精度的节点坐标

GRID* 双精度的节点坐标

CBAR 线元素

CTETRA，CTRIA3 四面体和三角形元素

CHEXA，CQUAD4，CPENTA 六面体，四边形和楔形元素

对于 2D 体网格，元素必须存在于一个恒定的 z 平面中。

（2）删除副节点

NASTRAN 可能在创建元素的过程中生成副的或重合节点。在写载入到 FLUENT 中的文件之前，这些节点必须从 NASTRAN 中删除。

PATRAN Neutral 文件

你可以使用三种不同的方法把一个 PATRAN Neutral 文件载入到 FLUENT 中。

你可以生成一个包含三角形，四边形，四面体，楔形或六面体元素的 PATRAN 表面或体网格，并使用 TGrid User’s Guide（TGrid 用户指南）中所述的命令把网格载入到 TGrid 中。如果有必要，还要完善网格的生成。

你可以生成一个线性的三角形，四边形，四面体，楔形，或六面体元素的 PATRAN 体单元，并使用 File/Import/PATRAN…菜单项直接载入网格。

你可以生成一个线性的三角形，四边形，四面体，楔形，或六面体元素的 PATRAN 体网格，然后使用过滤器 fe2ram 把 Neutral 文件转换成 FLUENT 使用的格式。关于把一个 PATRAN Neutral 格式的输入文件转换成一个 FLUENT 格式的输出文件的内容，在本节的后面有介绍。写完输出的文件之后，你可以使用 File/Read/Case…菜单项把文件读进 FLUENT。

在使用上面所述的方法 2 和 3 读进一个三角形和四面体的 PATRAN 体网格之后，你应

当执行平滑和交换命令以改进网格的质量。

（1）识别 PATRAN 数据集

下面的 PATRAN Neutral 文件包是通过 FLUENT 网格载入工具识别的。

Node Data 包类型 01

Element Data 包类型 02

Distributed Load Data 包类型 06

Node Temperature Data 包类型 10

Name Components 包类型 21

File Header 包类型 25

对于 2D 体网格，元素必须存在于一个恒定的 z 平面中。

（2）把元素分组以建立单元域

使用 **Named Component** 命令在 PATRAN 中对元素分组以建立多个单元域。在 FLUENT 中，所有被分组的元素一起被放在了一个单独的单元域中。如果元素没有被分组，那么 FLUENT 将把所有的单元放在一个单独的域中。

ANSYS 文件

你可以使用三种不同的方法把一个 PATRAN Neutral 文件载入到 FLUENT 中。

你可以使用 ANSYS 或 ARIES 生成一个包含三角形，四边形，四面体，楔形或六面体元素的表面或体网格，并使用 TGrid User's Guide（TGrid 用户指南）中所述的命令把网格载入到 TGrid 中。如果有必要，还要完善网格的生成。

你可以生成一个线性的三角形，四边形，四面体，楔形，或六面体元素的 ANSYS 体单元，并使用 File/Import/ANSYS…菜单项直接载入网格。

你可以生成一个线性的三角形，四边形，四面体，楔形，或六面体元素的 ANSYS 体网格，然后使用过滤器 fe2ram 把 Neutral 文件转换成 FLUENT 使用的格式。关于把一个 ANSYS 5.4 或 5.5 格式的输入文件转换成一个 FLUENT 格式的输出文件的内容，在本节的后面有介绍。写完输出的文件之后，你可以使用 File/Read/Case…菜单项把文件读进 FLUENT。

在使用上面所述的方法 2 和 3 读进一个三角形和四面体的体网格之后，你应当执行平滑和交换命令以改进网格的质量。

（1）识别 ANSYS 5.4 和 ANSYS 5.5 数据集

FLUENT 能够从 ANSYS 5.4 和 5.5（.cdb 文件）载入网格文件，并且保持文件中的最初的边界名程。下面的 ANSYS 文件的数据集是通过 FLUENT 网格载入工具识别的。

NBLOCK 节点块数据

EBLOCK 元素块数据

CMBLOCK 元素/节点块数据

元素必须是 STIF63 线性壳体元素。除此之外，如果没有显示的元素标识符的元素数据被使用，那么当建立域的时候，过滤器会假定元素的相继编号。

（2）使用 fe2ram 过滤器转换文件

如果你在把文件读入 FLUENT 之前选择手工转换 CAD 文件，你可以输入如下的命令：

utility fe2rm [dimension] format [zoning] input-file output-file

其中，方括号中的项表示可选项（不要输入方括号）。*dimension* 表示数据集的维数。-d2 取代 *dimension* 表示网格是二维的。对于一个 3D 网格，关于 *dimension* 不输入任何东西，因为 3D 是缺省值。

format 表示你所希望文件转换的形式。对于一个 ANSYS 文件，取代 *format* 的是 -tANSYS，对于一个 I-EDAS 文件则是-tIDEAS，对于一个 NASTRAN 文件是-tNASTRAN，对于一个 PATRAN 文件是-tPATRAN。要检查从其它 CAD 包中的转换是否已被添加，则键入 utility fe2ram –c1 -help 。

zoning 表示域是怎样在 CAD 包中鉴别的。*zoning* 被一个网格的-zID 取代，其中这个网格被特性标识符部分划分区域，或者被 zNONE 取代以忽略所有域的分组。对于一个被组划分区域的网格，关于 zoning，不需输入任何东西，因为通过组的分区制是缺省值。

Input-file 和 output-file 分别是最初的文件和你想把转换的网格信息写入的文件。

例如，如果你想把 2D I-DEAS 体网格文件 sample.unv 转换成一个名为 sample.grd 的输出文件，那么你将键入的命令是：

utility fe2ram –d2 –tIDEAS sample.unv sample.grd

4.3.7 FLUENT/UNS 和 RAMPANT Case 文件

如果你有一个 FLUENT/UNS 3 和 4 的 case 文件或一个 RAMPANT 2，3，或 4 的 case 文件，并且你想使用同一的网格运行一个 FLUENT 模拟，那么你可以使用 File/Read/Case…菜单项把相应的 case 文件读进 FLUENT 中。

4.3.8 FLUENT 4 Case 文件

如果你有一个 FLUENT 4 case 文件，并且你想使用同一的网格运行一个 FLUENT 模拟，那么使用 File/Import/FLUENT 4 Case…菜单项把文件载入 FLUENT。FLUENT 将从 FLUENT 4 case 文件中读网格信息和域的类型。

！注意，FLUENT 4 也许从 FLUENT 的当前释放中不同地通译一些压力边界。检查被 FLUENT 输出地转换信息以观察是否需要修改一些边界类型。

要手工地把一个 FLUENT 4 格式中的输入文件转换成一个当前的 FLUENT 格式中的输出文件，请键入命令：

utility fl42seg input-filename output-filename

写了输出文件之后，你可以使用 File/Read/Case…菜单项把这个文件读进 FLUENT。

4.3.9　FIDAP Neutral 文件

如果你有一个 FIDAP Neutral 文件，并且你想使用同一的网格运行一个 FLUENT 模拟，那么使用 File/Import/FIDAP…菜单项把文件载入 FLUENT。FLUENT 将从 FIDAP 文件中读网格信息和域的类型。

要手工地把一个 FIDAP 格式中的输入文件转换成一个 FLUENT 格式中的输出文件，请键入命令：

utility fe2ram [dimension] –tFIDAP7 input-filee output-file

其中，方括号中的项是可选的。（不要键入方括号。）对于一个 2D 文件，*dimension* 由-d2 取代。对于一个 3D 文件，关于 *dimension* 不输入任何东西，因为 3D 是缺省值。

写了输出文件之后，你可以使用 File/Read/Case…菜单项把这个文件读进 FLUENT。

5.3.10　读多个网格文件

也许存在一些这样的情况，你将需要读多个网格文件以形成你的计算域。下面列出了一些有关例子：

如果你打算解决一个多块网格，那么你可以在网格生成器中生成网格的每一块，并把它保存到一个分离的网格文件。

对于非常复杂的几何，作为一个分离的网格文件保存每一部分的网格可能更有效。

注意，不需要保证网格节点的位置与两个分离的网格会合处的边界上的节点相同一；FLUENT 会处理非正则网格界面。

读多个网格文件的过程如下：

在网格生成器中为整个域生成网格，然后保存每一单元域（或块或部分）到 FLUENT 一个分离的网格文件。

注意，如果你希望载入的一个（或多个）网格是结构网格（例如，一个 FLUENT 4 网格文件），你将首先需要使用 fl42seg 过滤器把它转换成 FLUENT 格式。

在启动求解器之前，使用 TGrid 或 tmerge 过滤器把网格文件合并成一个网格文件。TGrid 方法相对方便些，但是 tmerge 方法允许你在网格文件被合并之前对网格进行旋转，缩放或移动。

按照下面的步骤使用 TGrid：

把所有的网格文件读进 TGrid。当 TGrid 读网格文件时，它将自动地把它们合并成一个单独地网格。

保存合并后的网格文件。

按照下面的步骤使用 tmerge 过滤器：

对于 3D 问题，键入 utility tmerge -3d。对于 2D 问题，键入 utility tmerge -2d。

当输入命令提示符时，指定输入文件的名字和保存完整网格的输出文件的名字。对于每个输入文件，你可以指定缩放因子，移动距离，或一个旋转角。在下面的例子中，没有

缩放，移动或者旋转被执行。

```
    user@mymachine:>utility tmerge -2d
    Starting /Fluent.Inc/utility/tmerge2.1/ultra/tmerge_2d.2.1.13

    Append 2D grid files.
    tmerge2D Fluent Inc,Version 2.1.11
    Enter name of grid file (ENTER to continue) : my1.msh
    x, y scaling factor, eg. 1 1                : 1   1
    x, y translation, eg. 0 1                   : 0   0
    rotation angle (deg), eg. 45                :0
    Enter name of grid file (ENTER to continue) : <ENTER>
    Enter name of output file                   : final.msh

Reading…
    node zone: id 1, ib 1, ie 1677, typ 1
    node zone: id 2, ib 1678, ie 2169, typ 2
        .
        .
        .
done.
Writing…
    492 nodes, id 1, ib 1678, ie 2169, type 2.
    1677 nodes, id 2, ib 1, ie 1677, type 1.
        .
        .
        .
done.
Appending done.
```

在上面的例子中，其中没有对缩放，移动或者旋转有何要求，你可以简化输入如下：

utility tmerge -2d –c1 –p my1.msh my2.msh final.msh

以通常的方式把合并后的网格文件读进求解器（使用 File/Read/Case…菜单项）

对于一个正则网格，如果你不需要邻近单元域的一个边界，你可以使用 **Fuse Face Zones** 面板合并“重叠”的边界。匹配的面将被移到一个有内部（interior）的边界类型的新域。如果在每个初始域上的所有面已被移到了新域，那么初始域将不删除。

！如果你打算使用滑动网格，或你有邻近单元域间的非正则边界，你不应改合并重叠域。你必须用界面（interface）更换两个重叠域的类型。

4.4 非正则网格

在 FLUETN 中，可能会用到由非正则边界构成的网格。也就是说，在两个子域会合处的边界上网格节点的位置不需要同一。FLUENT 使用与滑动网格模型中所使用的相同的方法处理非正则网格，虽然这种情况下的网格不滑动。

4.4.1 非正则网格的计算

为计算通过非正则边界的通量，FLUENT 首先必须计算由边界构成的“界面”域间的交叉处。交叉处产生一个界面域，两个界面域在界面域处交迭（如图 4.4.1 所示）。如果其中的一个界面域延伸出了另一个界面域（如图 4.4.2 所示），FLUENT 将为不能交迭的两个界面域处的边界部分建立一两个附加的壁面域。

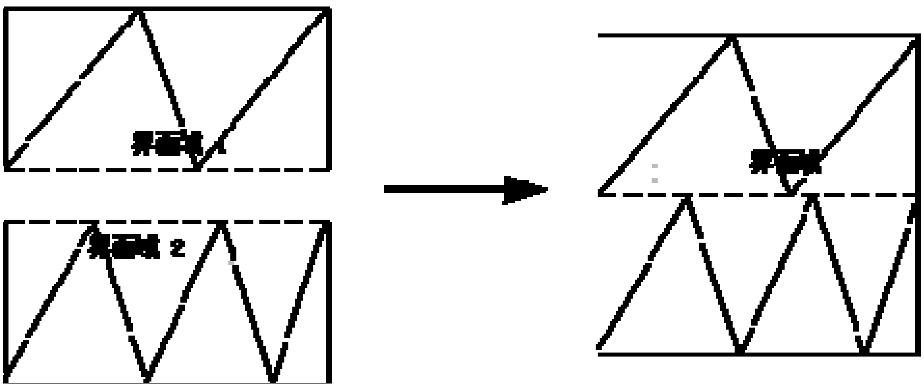


图 4.4.1：界面交叉处完全交迭的网格

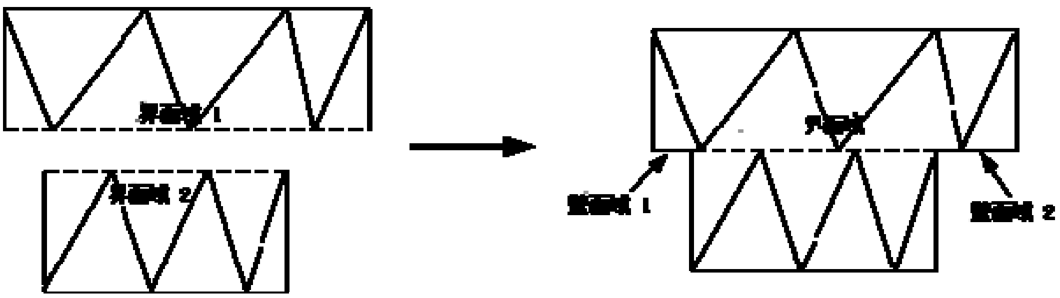


图 4.4.2：界面交叉处部分交迭的网格

主要地，过网格界面的通量是使用由两个界面域的交叉处所产生的面计算的，这优于从界面域的面本身计算。在如图 4.43 所示的例子中，界面域由面 A-B 和 B-B,和面 D-E 和面 E-F 构成。这些域的交叉处产生了面 a-d，d-b，b-e，和 e-c。在两个单元域交迭（a-d，d-b，b-e，和 e-c）处的域中产生的面组合起来形成了一个内部域。为计算过界面到单元 IV

内的通量，例如，不使用面 D-E，而使用面 d-b 和 b-e，分别把信息从单元 I 和 III 带进单元 IV 中。

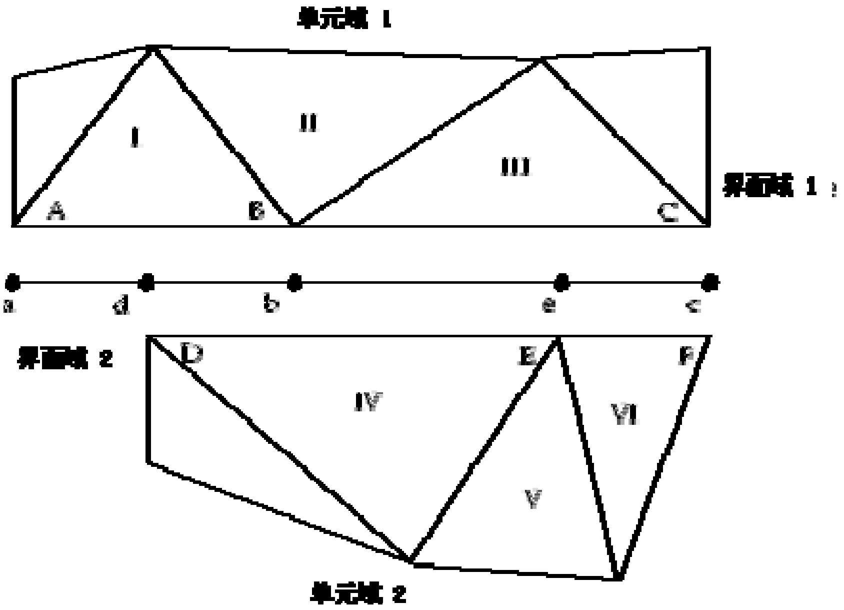


图 4.4.3：两维非正则网格界面

4.4.2 非正则网格的要求和局限

在使用非正则网格的时候，请注意以下的要求与限制：

网格界面可以是任何形状（包括 3D 中的非平面的表面），前提是两个界面边界基于同样的几何。如果网格中存在明显的特征（例如，90 度的角）或曲线，界面的两边接近地遵循这种特性是特别重要的。

例如，考虑两个同心圆定义带有一个圆的两个流体区域的情况，非正则界面在它们之间，如图 4.4.4 所示。因为与曲率的半径相比，在外部流动区域的界面边上的节点间距是粗糙的，所以界面不是很接近地遵循这个特性。

！一般来说，两个界面间地最大公差不应大于在那个位置上的它们的邻近单元尺寸；即在两个界面之间，不应有单元被完全地围住。

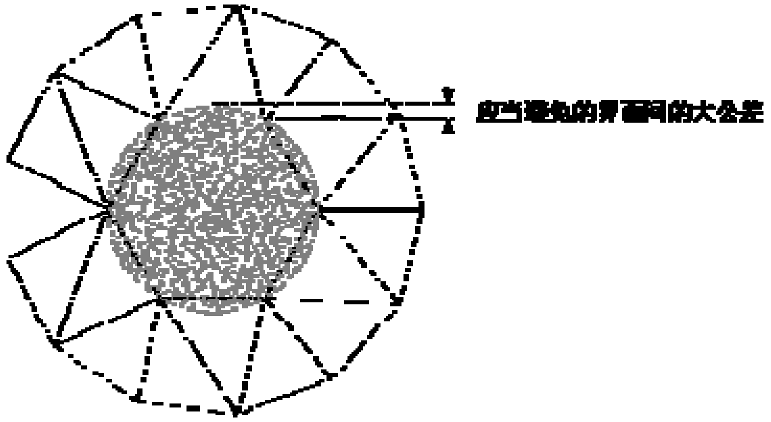


图 4.4.4：一个圆形的非正则界面

一个面域不能与多于一个的其它面域共享一个非正则界面。图 4.4.5 很好的说明了这一点。图中每一个体分别被划分了网格，在界面上没有配一个挨着一个的节点。为了创建一个非正则界面，你将需要使用三个表面：箱子的一个面（矩形 1： rectangle 1）和每一个管道的末端面（圆 1 和圆 2：circle 1 和 circle 2）。为在矩形 1 和两个圆之间创建界面，需要箱子的一个面（矩形 1： rectangle 1）和每一个管道的末端面之间的一一映射（图 4.4.6）。

如果你创建了一个被一非正则边界所分割的多单元域的单独网格，你必须确信每个单元域在非正则边界上有一个明显的面域。关于两个邻近单元域的面域要有相同的位置和形状。

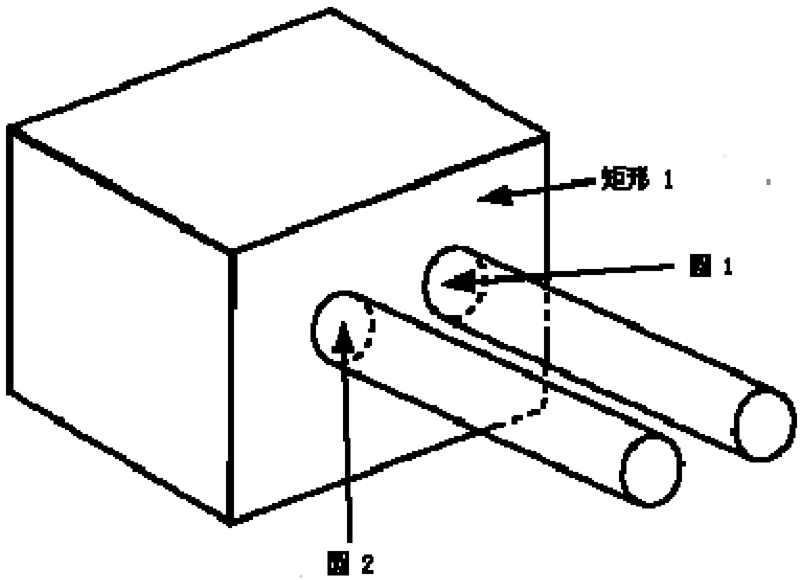


图 4.4.5：关于两个管道与箱子的一个面相交的图

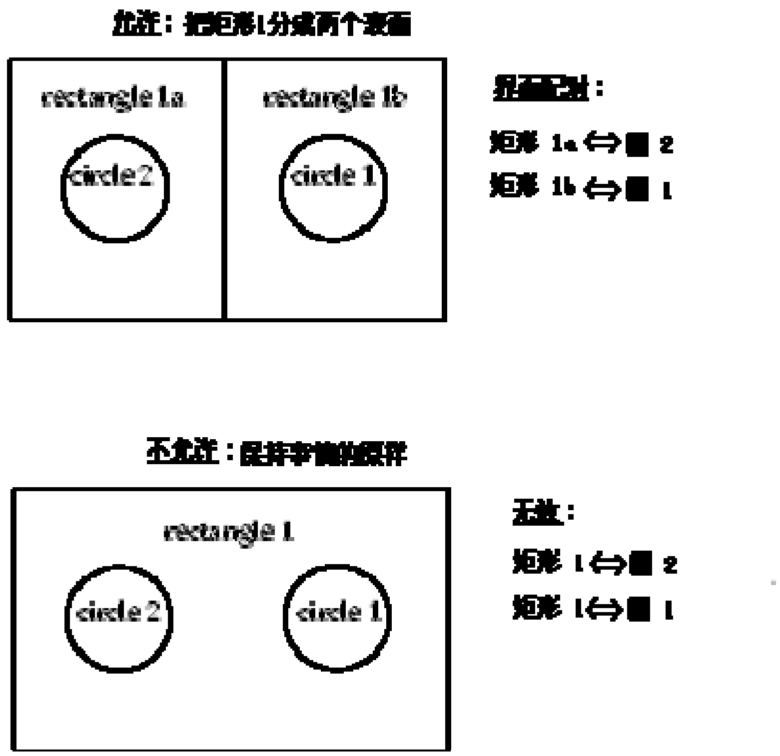


图 4.4.6：一一映射

在建立非正则界面前，所有周期域必须被正确地定位（或旋转或移动）。周期非正则界面必须正确地迭代；即，它们需要有相同的旋转或移动范围，另外，有同样的轴长。一般情况下，对于界面不是这样的，关于非重叠区域，一个壁面域在界面上被建立。

对于 3D 的情况，如果界面是周期的，仅有一对周期边界可以邻近界面。

第 4.4.4 节也介绍了有关使用非正则 FLUENT/UNS 和 RAMPANT 的情形。

4.4.3　在 FLUENT 中使用一个非正则网格

如果多域的网格包含非正则边界，那么必须遵循下面的步骤以保证 FLUENT 能够得到网格的一个解。

把网格读进 FLUENT。（若存在多个网格文件还没有被合并，则首先按照第 4.3.10 节的用法把它们合并成一个网格文件。）

在读了网格之后，改变每对由非正则边界的界面构成的域的类型。

Define→

Boundary Conditions…

在 **Grid Interfaces** 面板定义非正则网格界面（图 4.4.7）。

Define→
Grid Interfaces…

在面板中输入界面的名字。

分别在 **Interface Zone 1** 列和 **Interface Zone 2** 列中选择界面域来规定构成网格界面的两个界面域。! 如果其中的一个界面域比另一个小的多，那么应当规定小的域作为 **Interface Zone 1** 以改进交叉处计算的准确性。

若适当，设定 Interface Type。由两项：

对于周期问题，激活 Periodic。

如果界面处于一个固体域和一个流体域之间，或你愿意使用非正则界面在两个流体域之间建立一个耦合壁面，那么激活 Coupled。

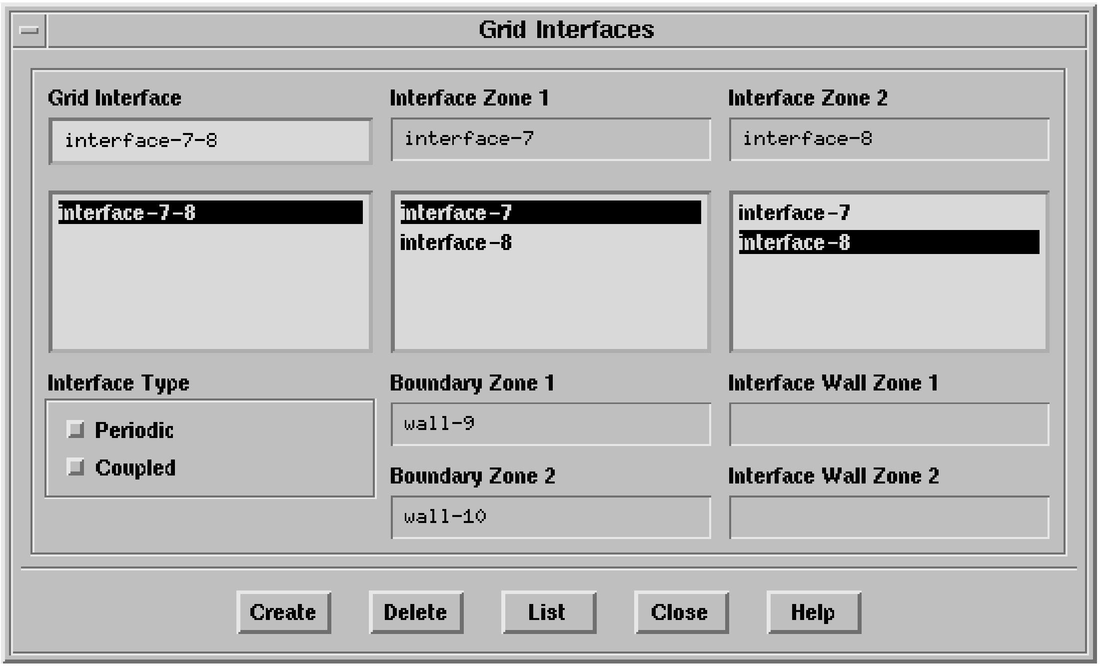


图 4.4.7：Grid Interfaces 面板

点击 **Create** 建立一个新网格界面。对于所有类型的界面，FLUENT 将为界面建立边界域（例如，wall-9，wall-10），这些边界域将显示在 **Boundary Zone 1** 和 **Boundary Zone 2**

的下面。如果你激活了 **Coupled** 选项，FLUENT 也将建立壁面界面域（例如，wall-4，wall-4-shadow），它们也将显示在 **Interface Wall Zone 1** 和 **Interface Wall Zone 2** 的下面。

如果两个界面域不完全交迭，检查为非交迭部分的边界所建立的域的边界域类型。如果域的类型不正确，可以使用 **Boundary Conditions** 面板改正。

如果你有任何耦合型的界面，通过在 **Boundary Conditions** 面板中更新界面的壁面域来定义边界条件。

Define →
Boundary Conditions…

如果建立了一个非正确的网格界面，那么可以在 **Grid Interface** 列中选择它，然后点击 **Delete** 按钮删掉它。（在界面建立的时候所建立的所有边界域或壁面界面域都将被删除。）

然后，你可以和通常一样继续进行问题的设置。

4.4.4 从一个 FLUENT/UNS 或 RAMPANT case 文件启动

有非正则界面的 FLUENT/UNS 和 RAMPANT case 文件可以被 FLUENT 不加改变的读入和使用。然而，也许你想重计算网格界面以利用 FLUENT 改良的交叉处的计算。你不能简单地删除原始的网格界面和重计算它。而是必须使用 define/grid-interfaces/recreate 文本命令。

Define →
recreate
grid-interfaces →

当选择这个命令时，FLUENT 将在域中重建所有的网格界面。然后可以通往常一样继续问题的设置或计算。

! 如果你已有了关于非正则情况的一个 FLUENT/UNS 或 RAMPANT 数据文件，你必须在使用 **recreate** 命令之前把它读进来。

4.5 检查网格

在 FLUENT 中，网格的检查能力包括域的范围，体积的数据统计，网格拓扑和周期边

界信息，单工计算器的确认，以及（轴对称的情况）关于 x 轴的节点位置的确认。通过在 Grid 下拉菜单中选择 Check 菜单项可以获得以上的相关信息。

Grid ⟶ Check

！一般来说，为了在开始设置问题前探察网格是否有问题，把网格文件读进求解器之后对网格进行检查是个很好的主意。

4.5.1 网格检查信息

当使用 Check 项的时候，FLUENT 产生的信息将会显示在控制台窗口中。输入的样本如下：

```
Grid Check

 Domain Extents:
   x-coordinate: min (m) = 0.000000e+00, max (m) = 6.400001e+01
   y-coordinate: min (m) = -4.538534e+00, max (m) = 6.400001e+01
 Volume statistics:
   minimum volume (m3): 2.353664e-05
   maximum volume (m3): 7.599501e-03
      total volume (m3): 2.341560e+00
   minimum 2d volume (m3): 4.027890e-04
   maximum 2d volume (m3): 1.230393e-03
 Face area statistics:
   mnimum face area (m2): 1.300719e－04
   maximum face area (m2): 3.781404e-02
 Checking number of nodes per cell.
 Checking number of faces per cell.
 Checking thread pointers.
 Checking number of cells per face.
 Checking face cells.
 Checking bridge faces.
 Checking right-handed cells.
 Checking face handedness.
 Checking for nodes that lie below the x-axis.
```

```
 Checking element type consistency.
 Checking boundary types:
 Checking face pairs.
 Checking periodic boundaries.
 Checking node count.
 Checking nosolve cell count.
 Checking nosolve face count.
 Checking face children.
 Checking cell children.
 Checking storage.
Done.
```

域的范围列出了 x，y 和 z 以米为单位的最大值和最小值。体积的数据统计包括以 m^3 位单位的单元体积的最大值，最小值和总的单元体积。最小体积值为负值意味着一个或多个单元有不合适的连通性。一个负体积的单元经常可以使用 **Iso-Value Adaption** 功能标记它们做适应及在图形窗口中查看它们，从而对这些单元加以确定。必须消除这些负体积才能继续进行流体计算。

拓扑的校正信息开始于每一单元的面和节点数。一个三角形单元（2D）应该有 3 个面和 3 个节点，一个四面体单元（3D）应该有 4 个面和 4 个节点，一个四边形单元（2D）应该有 4 个面和 4 个节点，一个六面体单元（3D）应该有 6 个面和 8 个节点。

关于每一个域的面的右手或左手法则应当被检查。域应当包含所有右手习惯的（向右旋的）面。通常情况下，一个有负体积的网格也将有左手习惯的面。直到排除这些连通性问题，才能获得一个流动解。

最后一个拓扑检定是检查元素类型的连贯性。如果一个网格不包含混合的元素（四边形和三角形或者六面体和四面体），FLUENT 将决定它没有必要知道元素的类型。这样做，它可以排除一些没有必要的工作。

对于轴对称的情况，x 轴下面的节点数目被列出来了。由于轴对称单元体积是通过旋转关于 x 轴的 2D 单元体而建立的，因此 x 轴下面的节点是被禁止的；所以，x 轴下面的节点会建立负体积。

对于有旋转的周期边界的解区域，最小值，最大值，平均值，以及指定的周期角被计算。一般的错误是不正确地规定角。对于有移动的周期边界的区域，边界信息被检查以确定边界真是周期的。

最后，对单工计算器做检定。节点、面的实际数目和求解器所构造的单元被与网格文件中的相应的头声明里所规定的值做了比较。任何差异都会被报告。

4.5.2 修复双重影子节点

如果 Grid/Check 包含中包含下面的信息：

WARNING：node on face thread 2 has multiple shadows.

那么它表示存在双重影子节点。这个错误仅发生在带有周期类型壁面的网格中。使用下面的文本命令修复这样的网格：

grid →

repair-periodic

modify-zones →

如果界面是旋转周期的，那么你会被提示输入旋转角。

4.6 报告网格的统计数据

在网格文件被读进 FLUENT 之后，存在报告关于网格的信息的几种方法。你可以报告当前问题所使用的内存总数，网格大小，及关于网格划分的数据统计。单元和面的域接域的合计数也可以被报告。从下面四小节介绍网格的数据统计：

4.6.1：网格的大小

4.6.2：内存的使用

4.6.3：网格域的信息

4.6.4：划分的数据统计

4.6.1 网格的大小

通过选择 Grid/Info/Size 菜单项可以输出网格中的节点、面、单元和划分的数目。

Grid →

Size

Info →

一个网格被划分以用于并行处理（请参阅第十六章），一个划分是这个网格中的一块。

输出的一个样例：

Grid Size

Level	Cells	Faces	Nodes	Partitions
0	7917	12247	4468	1

2 cell zones, 11 face zones.

如果你对单元和面是怎样在不同的域中分开的感兴趣，那么你可是用 Grid/Info/Zones 菜单项，如第 4.6.3 节所述。

如果你使用耦合显示求解器，那么关于每一网格水平的信息被输出。网格水平起因于为 FAS 多网格的收敛加速度建立粗糙的网格水平。输出的一个有关样例如下： Grid Size

Level	Cells	Faces	Nodes	Partitions
0	7917	12247	4468	1
1	1347	3658	0	1
2	392	1217	0	1
3	133	475	0	1
4	50	197	0	1
5	17	78	0	1

2 cell zones, 11 face zones.

4.6.2 内存的使用

FLUENT 有一个报告以下信息的性能：所使用的和分配的节点、面、单元、边及目标指示器的数目；所使用的和分配的排列内存的总数；求解器程序所使用的内存总数。

通过选择 Grid/Infl/Memory Usage 菜单项可以获得这个信息。

Grid ⟶

Memory Usage

Info ⟶

内存信息对于 UNIX 和 Windows 操作系统是不同的。在这里只介绍 Windows 系统下的内存信息。

关于 Windows 操作系统，处理内存信息的过程如下：

处理 Physical memory 是当前存在 RAM 中的指配堆内存。

处理 Virtual memory 是当前被交换到 Windows 系统记录文件中的指配堆内存。

处理 Total memory 是物理的和虚拟的内存之和。

注意以下几点：

内存信息不包括静态（代码）内存。

在 FLUENT 的一系列版本中，由于 Cortex 和求解器被包含在同一程序中，堆内存值包括求解器的存储（网格和解变量），和 Cortex（GUI 和图形内存）。

在并行版本中，Cortex 在它自己的程序中运行，所以堆内存值仅包括关于网格和解变量的存储。

使用 Task Manager（任务管理器）从关于 Windows 系统也可以得到更多关于 FLUENT 处理的资料。

4.6.3　网格域的信息

使用 Grid/Info/Zones 菜单项可以在控制台窗口输出关于每个域的节点、面和单元的信息。

Grid ⟶

Zones

Info ⟶

网格域的信息包括节点、每一面和单元域的总数，面或单元的数目，单元（在 3D 中，面）类型（三角形，四边形等），边界条件类型，域的标识符。输出的一个有关样例如下：

```
Zone sizes on domain 1:
    21280 hexahedral cells, zone 4.
        532 quadrilateral velocity-inlet faces, zone 1.
        532 quadrilatera pressure-outlet faces, zone 2.
1040 quadrilateral symmetry faces, zone 3.
1040 quadrilateral symmetry faces, zone 7.
    61708 quadrilateral interior faces, zone 5.
1120 quadrilateral wall faces, zone 6.
    23493 nodes.
```

4.6.4 划分的数据统计

选择 Grid/Info/Partitions 菜单项能够在控制台窗口输出网格划分的数据统计。

Grid ⟶
Partitions
Info ⟶

数据统计包括单元、面、界面和每一划分的邻域的数目。

4.7　修改网格

在网格被读进 FLUENT 之后，FLUENT 提供了几种修改网格的方法。你可以缩放或移动网格，合并或分开域，建立或分开周期域，以及合并边界。除此之外，你可以重新安排域中的单元以减少带宽。平滑和对角交换可以用来改进网格。

！无论何时修改网格，你都应该保存一个新的 case 文件（和一个 data 文件，如果 data 存在）。如果你想把一个旧的 data 文件读入，请确定 case 文件是否是初始的 case 文件，因为新的 case 文件可能与这个 data 文件不一致。

关于修改网格的内容将从以下 11 个方面进行介绍：

4.7.1：缩放网格
4.7.2：移动网格
4.7.3：合并（Merging）域
4.7.4：分离（Separating）域
4.7.5：建立周期域
4.7.6：分开（Slitting）周期域
4.7.7：合并（Fusing）面域
4.7.8：切开（Slitting）面域
4.7.9：挤出（Extruding）面域
4.7.10：重新安排计算域（domain）和地域（zones）
4.7.11：删除，使不活动和激活域

4.7.1　缩放网格

在内部，FLUENT 以米（SI 长度单位）为单位储存计算的网格。当网格被读进求解器

的时候，它被认为是以米为单位生成的。如果在建立网格时使用的是另一种长度单位（英寸，英尺，厘米等），那么你必须用米来标度网格的长度。为了做到这点，你可以从一列普通单位中选择某一单位以转换网格的单位，或者给出你自己定义的比例因子。每个节点坐标将会乘上相应的缩放比例因子。

标度也能够用来改变网格的物理尺寸。例如，在 x 方向设置比例因子为 2，在 y 和 z 方向上分别设置缩放比例因子为 1，这样可以把 x 方向的网格加长到原来的 2 倍。然而，由于缩放将改变网格中单元的纵横比，因此应当谨慎地运用各项异性的缩放。

！如果你打算缩放网格，那么应该在初始化流动或开始计算之前进行这个工作。否则缩放网格的任何数据都会无效。

你将使用 **Scale Grid** 面板（如图 4.7.1 所示）缩放一个标准度量单位的网格或运用自定义缩放比例因子。

Grid ⟶ Scale…

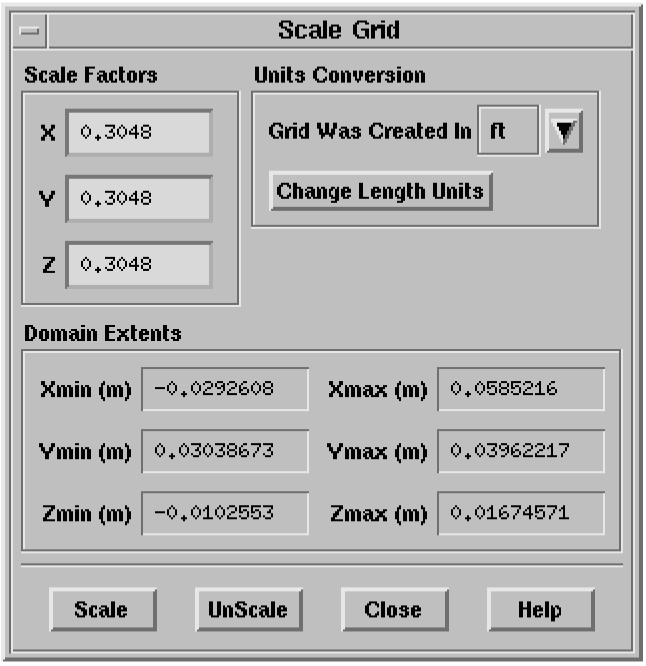


图 4.7.1：Scale Grid 面板

使用 **Scale Grid** 面板

缩放网格的步骤如下：

在 **Grid Was Created** 下面的框中选择适当的单位缩写：厘米，毫米，英寸或英尺，从而指出所建网格的单位。**Scale Factors** 将被自动地设置正确的值（例如，0.0254meters /inch 或 0.3048meters/foot）。如果建立网格使用的是除列在表中的其它单位，那么可以在 Scale Factors 中手工输入所需之值（例如，每码对应的米数）。

点击 **Scale** 按钮。**Domain Extents** 将被更新以显示正确的以米为单位的长度范围。如果你愿意在 FLUENT 中使用你原来的长度单位，那么可以按照下面叙述的步骤改变单位。

改变长度单位

如步骤 2 所述，当不改变单位而缩放网格时，若要把网格点的原来的尺寸从原来的单位转换成米，则乘上在 **Scale Factors** 中规定的每一节点坐标即可。如果打算使用原来的单位，而不是米，那么可以点击 **Chang Length Units** 按钮。这是在 **Set Units** 面板中改变长度单位的一个捷径。当点击 **Chang Length Units** 按钮时，**Domain Extents** 将被更新以显示原来单位的范围。这个单位将被用于所有以后输入的长度量。

不缩放网格

如果你使用了错误的缩放比例因子，不小心点击了 **Scale** 按钮两次，或者希望不缩放网格，那么你可以点击 **UnScale** 按钮。在 **Grid Was Created In** 列中选择 m，点击 **Scale** 不能不缩放网格。

改变网格的物理尺寸

也可以使用 **Scale Grid** 面板改变网格的物理尺寸。例如，如果网格时 5 英尺乘以 8 英尺，而你想把网格几何变为原来的二倍（10×16），那么你可以在 **Scale Factors** 下面的 X 和 Y 对应的框中以输入 2，然后点击 **Scale**。

4.7.2 移动网格

通过运用所有网格节点的 Cartesian 坐标的偏移量，可以“移动”网格。对于网格被设置为旋转轴不过原点的旋转问题，或者网格被设置为不以 x 轴为旋转轴的轴对称问题，移动网格是必要的。如果你想把原点移动到一个目标的特殊点（例如，一个浅盘的主要边）以便使得一个 XY 图有所需要的关于 x 轴的距离，移动网格也是有用的。

在 FLUENT 中使用 Translate Grid 面板（如图 4.7.2 所示）可以移动网格点。

Grid ⟶

Translate…

使用 Translate Grid 面板

移动网格的步骤如下：

在 **Translation Offsets** 下面 X，Y，和 Z 框中输入 x，y 和 z（对于 3D）方向的所要移动的值。你可以在当前的长度单位中指定正的或负的实型数。

点击 **Translate** 按钮。**Domain Extents** 将被更新并显示网格被移动的新范围。注意 **Domain Extents** 下面的内容纯粹是信息的显示，不能手工编辑。

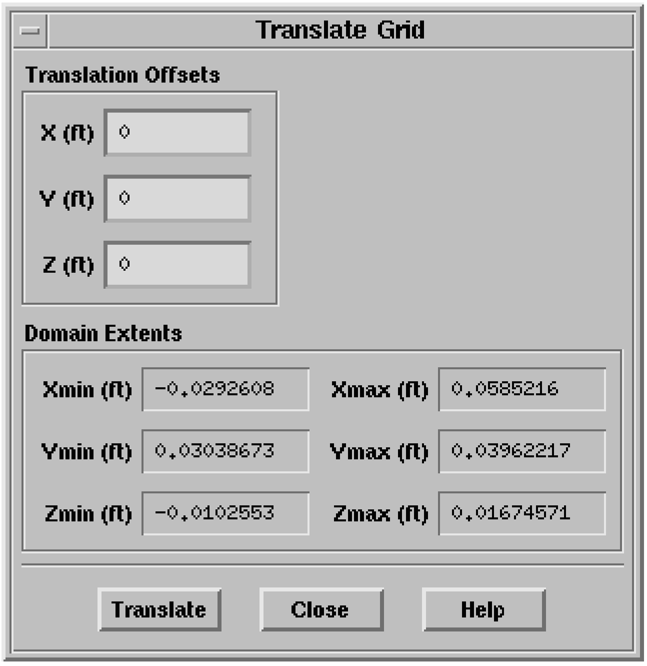


图 4.7.2：Translate Grid 面板

4.7.3 合并（Merging）域

合并域可以简化解过程。合并域包括把相似类型的多个域合并成一个域。在合并了相似域之后，设置边界条件及后处理可能会变地容易。如果合并域对数值分析的设置或后处理的效率不利，那么这样做也就没必要了。例如，对于一个大数目的域设置相同的边界条件参数会耗费时间，并可能带来一些矛盾。另外，数据的后处理经常涉及到使用域生成的表面。一个大数目的域经常转化成一个大数目的表面，对于展示选项（如，颜色等势线），

这些表面必须要加以选择。幸运地是，表面也可以被合并，最小化大数目的域对后处理的效率带来的负面影响。

在 Merge Zones 面板中执行域的合并。

Grid ⟶

Merge…

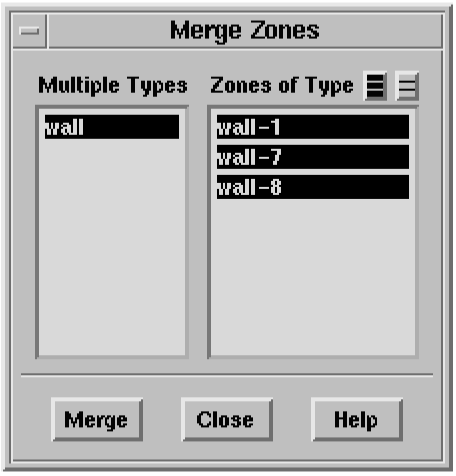


图 4.7.3：Merge Zones 面板

使用 Merge Zones 面板

同样类型的多个域合并成一个域的步骤如下：

在 **Multiple Types** 列表中选择域的类型。这个列中含有关于多个域的全部类型。一旦一个域从列表中被选择，相应的域则会显示在 **Zones of Type** 列中。

在 **Zones of Type** 列中选择两个或更多的域。

点击 **Merge** 按钮以合并所选的域。

！记得保存一个新的 case 文件（若 data 存在，则保存一个 data 文件）。

4.7.4 分离（Separating）域

在 FLUENT 中，把一个面或单元域分成相同类型的多个域存在着一些可行的方法。例如，在对一个管道生成网格时建立了一个壁面域，但你想对壁面上不同的部分规定不同的温度，那么可以根据自己的需要把这个壁面域分离成两个或多个壁面域。如果你计划使用

滑动网格模型或多个参考结构解决一个问题，但忘记了针对不同速度的移动域建立不同的流动域，那么你需要利用 FLUENT 提供的分离域的特性来把这个流动域分成两个或多个流动域。

分离面域的方法有四种，而分离单元域则有两种。周期域的分离将在第 4.7.6 节介绍。

注意，所有的分离方法允许你在执行分离的结果之前报告它。

分离面域

（1）分离面域的方法

对于锐角转角的几何，在重要角的基础上分离面域往往是比较容易的。如果面的法线矢量通过一个大于或等于所指定的重要角的角来区别，那么这些面将被放在不同的域。例如，如果你的网格由一个立方体构成，并且立方体的 6 个侧面都在一个壁面域里，那么你将指定一个89°的角为重要角。由于立方体的每个侧面的法线矢量通过90°区别于它相邻侧面的法线，因此 6 个侧面将被分别放在不同的壁面域中。

如果你有一个小面域，而你想把域中的每个面放进它自己的域，那么你可以基于面而分离面。

你也可以基于存储在适应性记录中的标记分离面域。例如，你可以在单元在区域中的位置的基础上对它们做适应标记。当你规定了哪一个记录被用于面域的分离时，做标记的所有单元的面将被放进一个新面域。（使用 **Manage Adaption Registers** 面板决定你愿意用的记录的标识符。）

最后，你可以基于邻近区域分离面域。例如，当你使用耦合壁面边界条件时，需要域上的面有一个一致的方位。一致方位仅能够在邻近区域上得到保证，所以你可能需要分离面域以允许适当的边界条件规定。

（2）分离面域的输入

基于角、面、使用标记、或区域分解一个面域，请使用 **Separate Face Zones** 面（如图 4.7.4 所示）。

Grid ⟶ Separate

Faces…

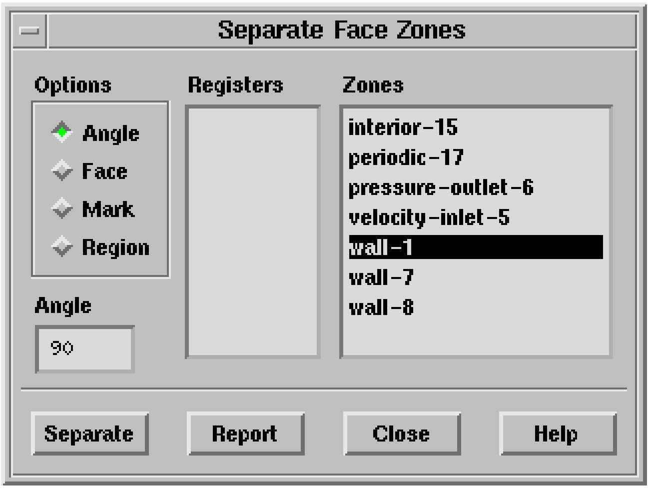


图 4.7.4：Separate Face Zones 面板

！如果你计划分离面域，你应当在使用（缺省的）悬挂节点适应方法执行任何适应之前进行这个工作。含有悬挂节点的面域不能分离。

分离面的步骤如下：

在 Options 下面选择分离方法（Angle，Face，Mark，或 Region）。

在 **Zones** 列中指定要被分离的面域。

如果你正在通过面或区域分离，跳过这一步。否则，继续下面过程中的一个。

如果你正在通过角分离面，那么在 **Angle** 框中规定重要角。

如果你正在通过标记分离面，那么在 **Registers** 列中选择要用的适应记录。

（可选择的）在你实际上分离面域之前要检查分离的结果，请点击 Report 按钮。报告的内容形式如下：

```
Zone not separated.
45 faces in contiguous region 0
30 faces in contiguous region 1
11 faces in contiguous region 2
14 faces in contiguous region 3
Separates zone 4 into 4 zone(s).
```

点击 Separate 按钮分离面域。FLUENT 将输出如下的信息：

```
45 faces in contiguous region 0
30 faces in contiguous region 1
11 faces in contiguous region 2
14 faces in contiguous region 3
Separates zone 4 into 4 zone(s).
Updating zone information…
  created zone wall-4 : 001
```

```
    created zone wall-4 : 002
    created zone wall-4 : 010
  done.
```

！当你通过适应标记分离面域时，有时你可能会发现一拐角单元的面将被放在错误的面域中。通常你可以基于角，借助执行另一个分离来移动犯规的面到一个新域，从而改正这个问题。然后，把这个新域与你想要被放置的面合并。

分离单元域

（1）分离单元域的方法

如果你有两个或更多个封闭的单元域，它们有共同的内边（如图 4.7.5 所示），但是所有的单元包含在一个单元域中，那么你可以使用域分离的方法把单元分成明显的域。注意，如果共享内边界是 interior 类型，那么在执行分离之前必须把它变化成另一种双边的面域类型（fan，radiator 等）。

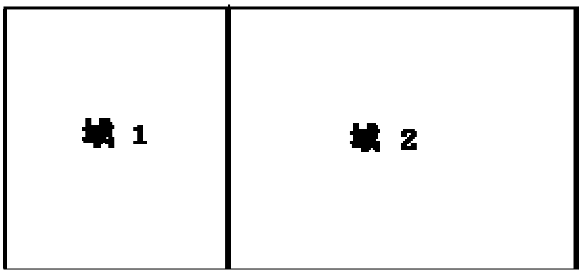


图 4.7.5：基于区域的单元域分离

你也可以基于储存在适应记录里的标记分离单元域。你可以使用任何适应方法标记关于适应的单元。当指定了哪一个记录被用于单元域的分离时，被标记的单元将被放进一个新的单元域。（使用 **Manage Adaption Registers** 面板决定你所希望使用的记录的标识符。）

分离单元域的输入

要基于区域或适应标记分解一个单元域，请使用 Separate Cell Zones 面板（如图 4.7.6 所示）。

Grid ⟶ Separate

Cells…

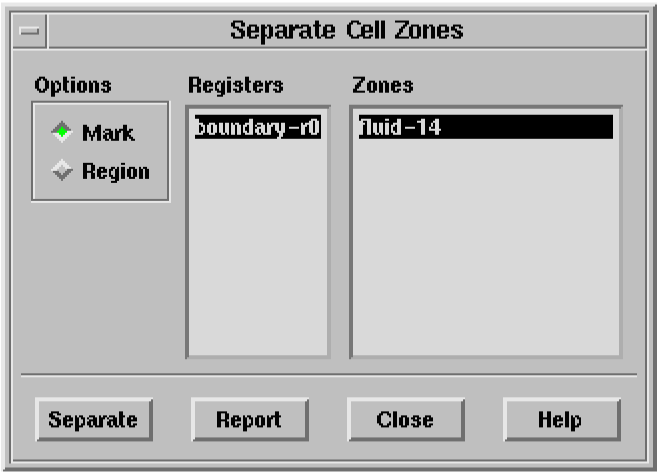


图 4.7.6：Separate Cell Zones 面板

！如果你计划分离单元域，你应当在使用（缺省的）悬挂节点适应方法执行任何适应之前进行这个工作。含有悬挂节点的单元域不能分离。

分离单元的步骤如下：

在 Options 下面选择分离方法（Mark，或 Region）。

在 **Zones** 列中指定要被分离的单元域。

如果你借助标记分离，那么在 **Registers** 列中选择要用的适应记录。

（可选择的）在你实际分离单元域之前要检查分离的结果，请点击 **Report** 按钮。报告的内容形式如下：

```
Zone not separated.
Separates zone 14 into two zone(s),with 1275 and 32 cells.
```

点击 Separate 按钮分离单元域。FLUENT 将输出如下的信息：

```
Separates zone14 into two zones, with 1275 and 32 cells.
No faces marked on thread, 2
No faces marked on thread, 3
No faces marked on thread, 1
No faces marked on thread, 5
No faces marked on thread, 7
No faces marked on thread, 8
No faces marked on thread, 9
No faces marked on thread,61
Separates zone 62 into two zones, with 1763 and 58 faces.
All faces marked on thread, 4
No faces marked on thread, 66
Moved 32 cells from cell zone 14 to zone 10
```

```
    created zone interior-4
    created zone interior-6
    created zone fluid-14 : 010
  done.
```

如上面例子所示，一个单元域的分离经常也会导致面域的分离。当你通过标记分离时，被移到一个新域的单元，它们的面将被放进一个新的面域。而当你通过区域分离时，没必要如此。如果你发现任何面放置错误，请看上面所述的面域分离的有关内容。

4.7.5 建立周期域

FLUENT 允许使用正则或非正则的周期域建立周期边界。通过耦合一对面域可以对网格设置周期。若两个域有同样的节点和面分布，则可以建立一个正则周期域。若边界上两个域在一个规定的公差内不相同，则可以建立一个非正则周期域。

！在创建或分开一个周期边界后记得保存一个新的 case 文件（若 data 存在，还要保存一个 data 文件）。

建立正则周期域

使用 make-periodic 文本命令建立正则周期边界。

Grid →
Modify-zones →
Make-periodic

你将需要规定构成周期对的两个面域，方法是输入它们的全名或它们的标识符；指出它们是旋转周期还是移动周期。关于规定周期域和它的匹配影子域的顺序是不重要的。

```
/grid/modify-zones>mp
Periodic zone [()] 1
Shadow zone [()] 4
Rotational periodic? (if no, translational) [yes] n
Create periodic zones? [yes] yes

  computed translation deltas: -2.000000 -2.000000
  all 10 faces matched for zones 1 and 4.
```

```
zone 4 deleted
Created periodic zones.
```

当创建了一个正则周期边界时，求解器将检查在所选域上的面是否“匹配”（即与面相应的节点是否一致。）关于一个面的匹配公差是这个面的最小边长的一部分。如果创建周期边界失败，那么使用 matching-tolerance 命令改变匹配公差，然而其值不应超过 0.5 或可能造成周期域配合错误以及使网格恶化。

Grid ⟶
Modify-zones ⟶
Matching-tolerance

建立非正则周期域

要建立非正则周期边界，需要把非正则周期域改变成界面域。邻近单元域的原点和轴都需要设置。例如，如果 interface-15 和 interface-2 是两个非正则周期域，那么使用 define/grid-interfaces 文本菜单中的 make-periodic 命令设置非正则周期边界。

Define ⟶
Grid-interfaces
Make-periodic…

例如：

```
/define/grid-interfaces> make-periodic
Periodic zone [()] interface-15
Shadow zone [()] interface-2
Rotational periodic? (if no, translational) [yes] yes
Rotation angle (deg) [0] 40.0
Create periodic zone? [yes] yes
Grid-interface name [] fan-periodic
```

4.7.6　分开（Slitting）周期域

使用 slit-periodic 命令可以分开（解藕）一个周期对中的域。

Grid ——→
Modify-zones
Slit-periodic

你将规定周期域的名字和标识符，并且求解器将解耦对中的两个域（周期域和它的影子）并把它们变成两个对称域：

```
/grid/modify-zones> sp
periodic zone [()] periodic-1
  Separated periodic zone.
```

4.7.7　合并（Fusing）面域

通过集合多个网格域来合并边界（和合并双重节点和面）。当域被分成子域，及对每个子域网格是分开生成的时，在把网格读进求解器之前要把子域合并一个文件中（详细内容请看第 4.3.10 节）。这种情形发生在分开地生成一个多块网格中每一块并分别的保存到一个单独的网格文件中。另一种情形是在网格生成中把关于一个复杂几何的每一部分网格保存为一个分开部分的文件。（注意，在两个子域交会的边界处不需要网格节点位置同一；详细内容请看第 4.4 节。）

Fuse Face Zones 面板（图 4.7.7）允许你合并双重节点和删除这些人工内边界。

Grid ——→
Fuse…

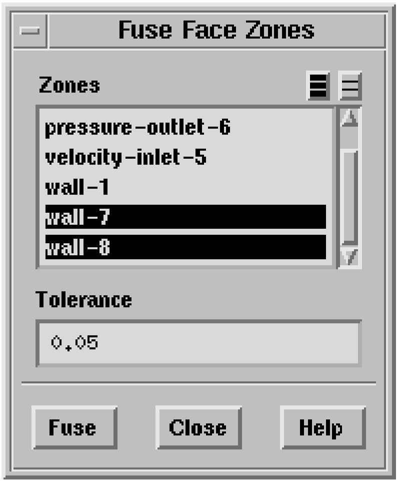


图 4.7.7：Fuse Face Zones 面板

如第 4.3.10 节所述，当网格文件被合并时，双重节点所处的边界被安置了域 ID 数（类似于其它边界）。当 tmerge 或 TGrid 报告域 ID 数时，或者在全网格被读进，展示所有边界网格域和使用鼠标探测器按钮决定域名之后，需要知道域 ID 数。

（1）合并面域的输入

合并面域的步骤如下：

在 **Zones** 列中选择要合并的域。

点击 **Fuse** 按钮合并所选域。

如果使用 **Tolerance** 的缺省值没有使所有适当的面合并，那么增加 **Tolerance** 的值，然后再试着合并域。（这里的公差（tolerance）和 4.7.5 节所述的匹配公差相同。）**Tolerance** 不应超过 0.5，否则可能会合并错误的节点。

！记得在合并面之后保存一个新的 case 文件（若 data 存在，保存一个 data 文件）。

（2）合并分支部分的域

从结构网格生成器或求解器（例如 FLUENT 4）导入的网格往往可以是 O-类型或 C-类型的带有一个凹进分部的网格，其中这个凹进部分处有一致双重节点位于一个周期边界上。由于 FLUENT 使用一个非结构网格表示，保留这个人工内边界是没有原因的。

要合并这个周期边界，首先必须分开这个周期域，如 4.7.6 节所述。这将建立能够使用上面的步骤合并的两个对称域。

注意，如果需要合并一个非周期域的部分，那么必须使用 **fuse-face-zones** 命令。

Grid ⟶

Modify-zones

Fuse-face-zones

这个命令将提示你输入名字或要被合并的每个域的 ID。（需要输入相同的域两次。）要改变节点公差，请使用 **matching-tolerance** 命令。

4.7.8　分开（sliting）面域

面域分开（或切开）的功能有两个：

可以把任何双边类型的一个边界域分成两个明显的域。

可以把一个耦合壁面域分成两个明显的、非耦合的壁面域。

当分开一个面域时，求解器将复制除了那些位于域的尽头（2D）或边上（3D）的节点外的所有面和节点。节点和面的一个设置将属于所得边界面中的一个，其它设置则属于其它域。关于每一尽头的共享点，其唯一不好的结果是当以图形的形式展示一个切开边界的解数据时，可能在这些点处出现一些不正确的解。（注意，如果在切开之后改编边界，那么不能把边界再合并回来。）

通常情况下，没必要手工切开一个面域。有两条边的壁面被自动地切开，但仍是耦合的。（这个耦合仅对网格而言，独立于热耦合。）适应程序把这些耦合壁面作为周期边界；对一个壁面做适应也会使它的影子产生相同的适应。要对一个壁面做独立于它的影子的适应，应当切开耦合壁面以得到两个明显的壁面。

！不应把“分开（slitting）”和“分离（separating）”混淆。当分开一个面域时，另一些面和节点被建立和放置在一个新域中。而当分离一个面域时，一个新域将被建立，但没有新面或节点被建立；只是现有的面和节点在域中被重新分布。

分开面域的输入

使用命令 slit-face-zone 可以分开一个面域。

Grid ⟶

Slit-face-zone
Modify-zones

你将规定面域地名字或 ID，求解器将用两个域替代这个域：

```
/grid/modify-zones> slfz
face zone id/name [] wall-4


  zone 4 deleted
  face zone 4 created
  face zone 10 created
```

！记得在分开面后保存一个新的 case 文件（若 data 存在，还要保存一个 data 文件）。

4.7.9　挤出（Extruding）面域

挤出一个边界面域使得不必退出求解器就能扩大解域。关于挤出功能的一个典型应用是当再循环流动碰撞到一个流动出口时扩大解域。在平均面的法线到面域的节点所得的面和法线矢量的形状基础上，当前挤出功能创建了柱状或六面体层。通过规定一列位移（在 SI 单位中）或一个总距离（在 SI 单位中）和参量坐标可以定义挤出过程。

通过位移长度规定挤出

使用 extude-face-zone-delta 命令，通过输入一列位移长度（在 SI 单位中）可以规定挤压。

Grid ⟶

extude-face-zone-delta
Modify-zones⟶

你将被提示输入边界面域的 ID 和名字及一列位移长度。

通过参量坐标规定挤压

使用 extude-face-zone-para 命令，通过规定一个距离（在 SI 单位中）和参量坐标可以规定挤压。

Grid ⟶

extude-face-zone-para
Modify-zones⟶

你将被提示输入边界面域的 ID 和名字，一个总距离，及一列参量坐标。这列参量坐标应当以 0.0 开始并到 1.0 结束。例如，下面这列参量坐标将建立两个相等的空间挤压层：0.0，0.5，1.0。

4.7.10　重新安排计算域（domain）和地域（zones）

通过重新安排内存中的节点、面和单元可以重新给计算域排序以提高求解器的计算性能。Grid/Reorder 子菜单含有记录计算域和地域的命令，及输出目前网格划分带宽的命令。计算域被重新安排以提高内存存取的效率，为了用户界面的便利，地域可以被重新排序。带宽给出了单元在地域和内存中的分布的情况。

选择 Domain 菜单项重新安排计算域。

Grid ⟶ Reorder ⟶ Domain

选择 Zones 菜单项重新安排地域。

Grid ⟶ Reorder ⟶ Zones

最后，选择 Print Bandwidth 菜单项输出目前网格划分地带宽。这个命令输出关于每一网格划分的半带宽和最大内存长度。

Grid ⟶ Reorder ⟶ Print Bandwidth

例如，使用计算域重新排序所产生的一个典型输出如下：

```
>> Reordering domain using Reverse Cuthill－McKee method:
      zones, cells, faces, done
   Bandwidth reduction = 809/21 = 38.52
   Done
```

如果输入了一个带宽，有一个相似的报告如下：

```
Masimum cell distance = 21
```

！记得在重新安排后保存一个新的 case 文件（若 data 存在，还要保存一个 data 文件）。

4.7.11　删除，使不活动和激活域

你可以从你的 FLUENT case 文件中永久地删除一个单元域和所有相关的面域，或永久地使域不活动。

要永久的删除域，请在 Delete Cell Zones 面板中选择它们（图 4.7.8），然后点击 Delete。

Grid ⟶
Delete…

Zone ⟶

要使域不活动，请在 Deactivate Cell Zones 面板（图 4.7.9）中选择它们，然后点击 Deactivate。

Grid ⟶
Deactivate…

Zone ⟶

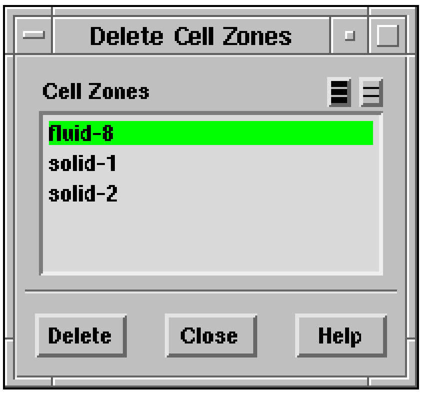


图 4.7.8：Delete Cell Zones 面板

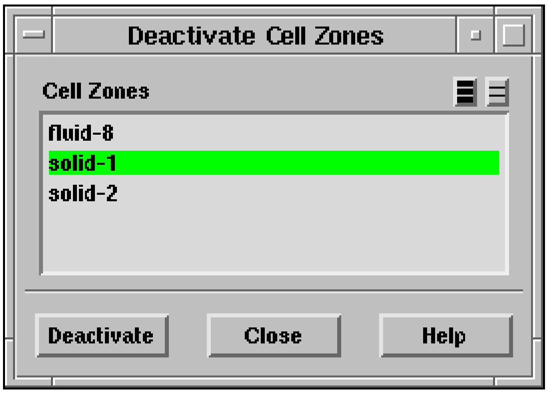


图 4.7.9：Deactivate Cell Zones 面板

当使用 Deactivate Cell Zones 面板使一个域不活动时，FLUENT 将从网格中或从所有相关的求解器循环中移走域。那么你能够使域不活动及使用 Activate Cell Zones 面板（图 4.7.10）恢复最后可用于域的数据。

Grid ⟶
Activate…

Zone ⟶

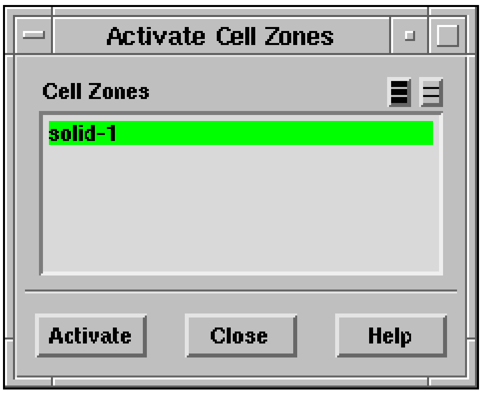


图 4.7.10：Activate Cell Zones 面板

注意，这个 Activate Cell Zones 面板仅与前面使不活动的域组合使用。

！域的删除，使不活动，和激活仅适用于串行情况，而不使用于并行情况。使不活动将把所有相联系的内部面域（即，风扇，内部，多孔跳跃，或辐射体）分离成壁面和壁面影子对。在使不活动之后，你需要确定关于壁面和壁面影子对的边界条件被正确地恢复到

使域不活动之前的设置状态。值得注意的还有，如果你打算以后在使它们不活动，那么需要确定在使不活动过程中分离的面域没被修改。然而，这里适应是被支持的。

4.8　将网格分区用于并行计算

在 FLUENT 中使用并行求解器时，需要把网格划分或细分成在分离处理机上能被求解的单元组（如图 8.4.1）。划分网格有两种方法，一种是，当你读进一个未划分的网格到并行求解器时，可以使用自动划分运算法则；另一种是，在串行求解器中或把一个网格读进并行求解器之后自己执行划分。划分网格在建立问题（定义模型，边界条件等）之前或之后均可，虽然最好是在建立问题之后。

注意，除了非正则界面出现，计算节点中的单元的相对分布在网格适应时将保持，所以适应之后在划分网格则不需要。

在划分之前使用串行求解器建立问题，执行任务的机器必须有足够的空间以读进网格。如果网格太大而不能读进串行求解器，那么你可以直接把未划分的网格读进并行求解器。在这种情况下，你将在初始划分之后再建立问题。若必要，你将能够对网格手工进行再划分。

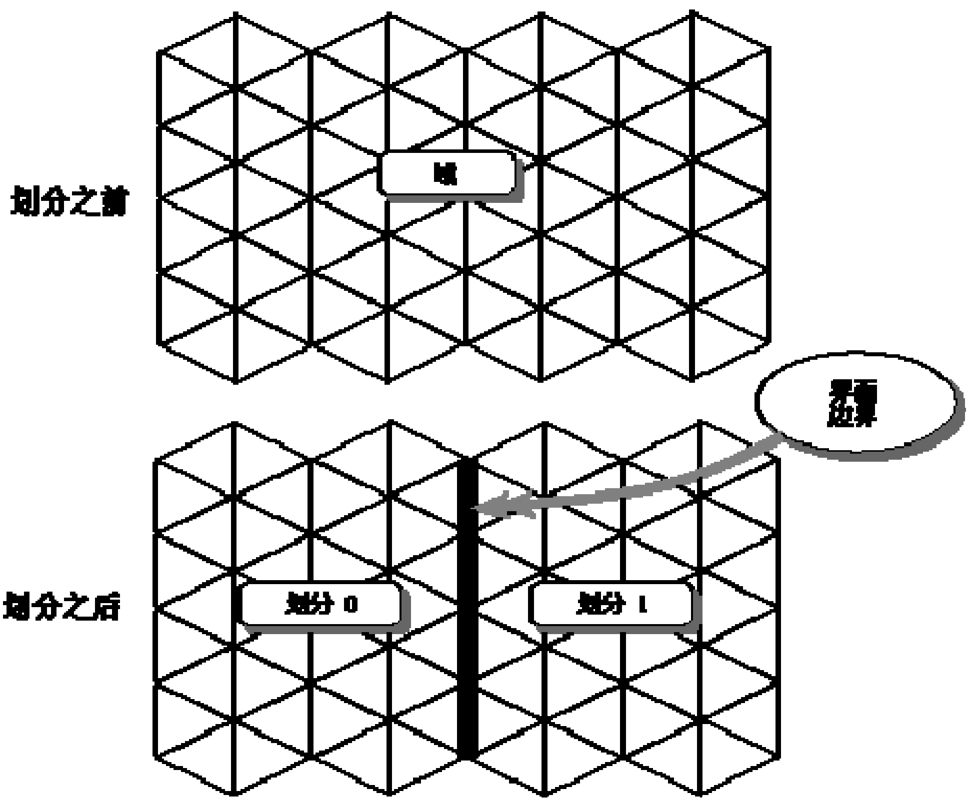


图 4.8.1：划分网格

4.8.1　自动地划分网格

关于自动的网格划分，在把一个 case 文件读进并行版本的求解器之前，你可以选择对分方法及关于建立网格划分的其它选项。对于一些方法，你可以执行预测验以保证尽可能使用最好的方法。

如果 case 文件包含滑动网格，或非正则界面，其中你打算在计算过程中在这个界面上执行适应，那么你必须在串行求解器上划分网格，然后再把它读进并行求解器。

在并行求解器中自动划分的步骤如下：

1．（可选）在 Auto Partition Grid 面板中设置划分参数（如图 4.8.2）。

Parallel⟶

Auto Partition…

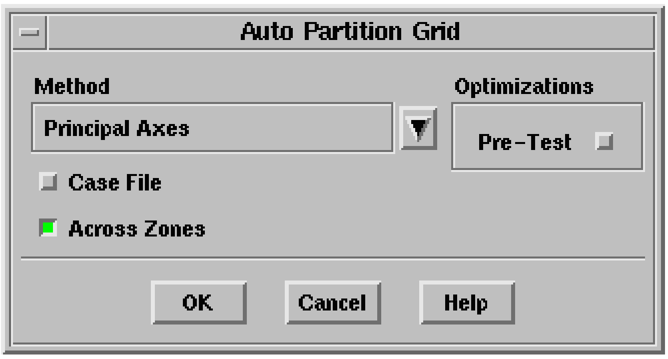


图 4.8.2：Auto Partition Grid 面板

如果你正在读进一个 mesh 文件或关于没有划分信息可用的一个 case 文件，并且保持打开的 Case File 选项，那么 FLUENT 将使用显示在 Method 下面列中的方法划分网格。

如果你想指定划分方法和你自己的相联系的选项，步骤如下：

关掉 **Case File** 选项。面板中的其它选项变地可用。

在 Method 下面的列中选择对分方法，选择的技巧在第 4.8.3 节有介绍。

你可以选择独立地应用每一单元域，或使用 **Across Zones** 检查按钮允许划分通过域的边界。除非在解相过程中不同域中的单元十分需要不同的计算数量（例如，域包含固体的和流体的两个域），否则不建议单独地划分单元域（通过关掉 Across Zones 检查按钮）。

如果你已经选择了 Principal Axes 或 Cartesian Axes 方法，那么为改进划分，你可以在实际划分发生之前通过激活不同的对分指导的自动测验。

点击 OK。

如果你读进的case文件中已有划分的网格，及划分的数目最终分成了计算节点的数目，那么你可以在Auto Partition Grid面板中保留Case File的缺省选择。这指导FLUENT使用case文件里的划分。

2．读case文件。

File ⟶

Case…

Read ⟶

在自动划分过程中的报告

当网格被自动划分时，关于划分程序的一些信息将在控制台窗口中输出。如果你想要另外的信息，你可以在划分完成后从Partition Grid面板中输出一个报告。

Parallel ⟶

Partition…

当你点击**Partition Grid**面板中的**Print Active Partitions**或**Print Stored Partitions**按钮时，FLUENT将在控制台窗口输出划分ID，单元、面和界面数，及界面与每一激活的或存储划分的面的比率。另外，还会输出单元、面、界面和面比率变量的最小值和最大值。

4.8.2 手工划分网格

划分网格的指导方针

推荐使用下面的步骤手工划分一个网格如下：

使用缺省的对分方法（Principal Axes）和最优化（Smooth）划分网格。

检查划分的数据统计。你的目标是在保持一个平稳的负荷（Cell variation）时获得界面比率的变量（Interface ratio variation）和全局界面比率（Global interface ratio）的小值。如果数据统计不可接受，尝试其它的对分方法。

一旦对你的问题决定了最好的对分方法，你可以打开**Pre-Test**以进一步改进它。

如果决定了，你也可以使用**Merge**最优化来改进划分。

使用 Partition Grid 面板

对于网格划分，你需要为建立网格划分而选择对分方法，设置划分的个数，选择域或记录，并且选择要用的最优化。对于一些方法，你也可以执行预测验以保证尽可能使用最佳的对分方法。一旦在 **Partition Grid** 面板中设置了所有符合自己要求的参数，使用前面所说的方法和优化法把网格细分成所选的划分个数。

你可以按照下面的方式在 **Partition Grid** 面板（图 4.8.3，在并行求解器中；图 4.8.4，在串行求解器中）中设置相关的输入。

Parallel ⟶

Partition…

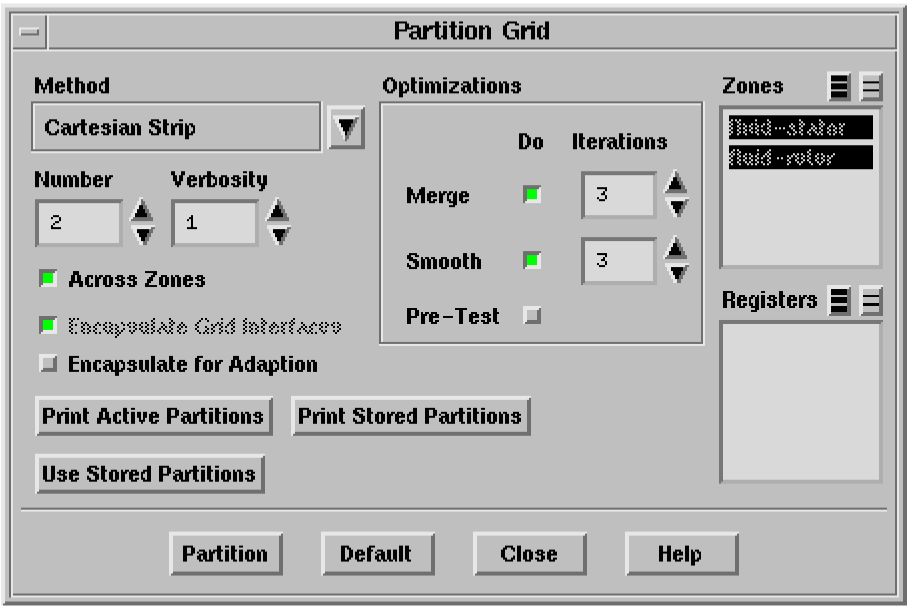


图 4.8.3：在并行求解器中的 Partition Grid 面板

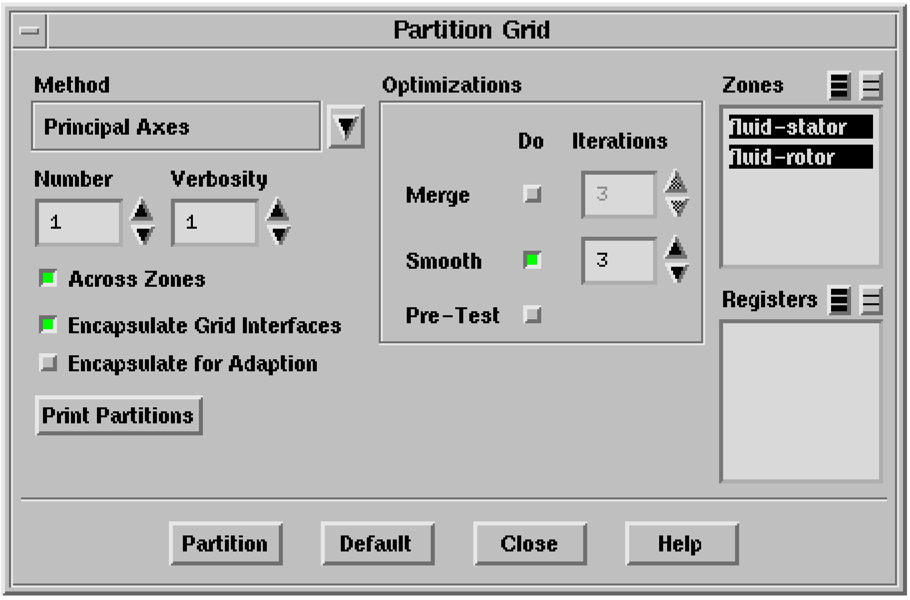


图 4.8.4：在串行求解器中的 Partition Grid 面板

在 **Method** 下面的列中选择对分方法。选择的技巧在第 4.8.3 节中有介绍。

在 **Number** 整数框中选择的网格所要划分的个数。网格划分的个数必须是用于并行计算的处理机数目的整数倍。

你可以选择独立地应用到每一单元域的划分，或者使用 **Across Zones** 检查按钮允许划分通过域的边界。除非在解相过程中不同域中的单元十分需要不同的计算数量（例如，域包含固体的和流体的两个域），否则不建议单独地划分单元域（通过关掉 **Across Zones** 检查按钮）。

在计算过程中，如果你喜欢网格中的周围环绕所有非正则网格的单元位于一个单独的划分中，那么你可以选择 **Encapsulate Grid Interfaces**。如果你的 case 文件中包含非正则界面，其中你计划在计算过程中在这个界面上执行适应，那么你必须在 **Encapsulate Grid Interfaces** 和 **Encapsulate for Adaption** 选项被打开的串行求解器上划分网格。

如果你在串行求解器上激活了 **Encapsulate Grid Interfaces** 选项，**Encapsulate for Adaption** 选项也将可用。当你选择了这个选项，单元的附加层被包封以便在并行适应过程中没必要移动单元。

你可以使用 **Optimizations** 下面的项激活和控制想要用的最优化方法。打开 Do check 按钮，你可以激活 **Merge** 和 **Smooth** 方案。对每一种方案，你也可以设置 **Iterations** 的数目。每个最优化方案将被应用直到达到适当的标准，或执行了迭代的最大数。如果 Iterations 计算器被设置为 0，那么最优化项将被应用直到完成，迭代的最大数没有限制。

如果你选择了 Principal Axes 或 Cartesian Axes 方法，你可以在实际划分发生之前通过激活不同对分指示的自动测验改进划分。要使用预测验，请打开 Pre-Test 选项。

在 Zones 或 Registers 列中，选择你想划分的域或记录。对于大多数情况，你将选择划分整个域的所 Zones（缺省）。

点击 **Partition** 按钮划分网格。

如果你决定新划分优于前面的划分（如果网格已经划分了），点击 Use Stored Partitions 按钮使新储存的单元划分成为活动的单元划分。当保存一个 case 文件时存储的单元划分被使用，活动的单元划分用于当前的计算。

当在并行模拟中使用动态网格模型，**Partition** 面板包含一个 **Auto Repartition** 选项和一个 **Repartition Intrval** 设置。缺省时，**Auto Repartition** 选项被选择，这时，FLUENT 基于各种模拟参数自动地决定最适当的再划分间隔。有时，使用 **Auto Repartition** 选项会提供不重要的结果，所以可以使用 **Repartition Interval** 设置。当一个重新划分被实施时，**Repartition Interval** 设置让你规定间距（分别在时间步和迭代里）。当不希望重新划分时，你可以设 **Repartition Interval** 为 0。

！注意，当动态网格和局部重新分网格被应用时，并行 FLUENT 中的更新网格与串行 FLUENT 相比可能稍有不同，即解中有很小的差别。

在手工划分过程中的报告

当网格被划分时，划分程序的信息将在控制台窗口中输出。作为默认，求解器将输出所建划分的个数，执行的对分个数，划分所需时间，及单元、面、界面和面比率变量的最小值和最大值。如果 Verbosity 从缺省值 1 增加到 2，所用的划分方法，划分的 ID，单元、面和界面的个数，及界面与每一划分面的比率也将在控制台窗口输出。如果 Verbosity 减到 0，仅报告建立的划分个数和划分所需的时间。

在划分完成后，你可以要求只输出一部分报告内容。在并行求解器中，当你点击 **Partition Grid** 面板中的 **Print Active Partitions** 或 **Print Stored Partitions** 按钮时，FLUENT 将在控制台窗口输出划分的 ID，单元、面和界面数，及界面与每一激活的或存储划分的面的比率。另外，还会输出单元、面、界面和面比率变量的最小值和最大值。在串行求解器中，当你点击 **Print Partitions** 时，你将获得关于存储划分的相同的信息。

！当应用并行版本的 FLUENT 时或选择了作为划分 Method 的 Metis 时，在域或记录中划分网格是没有用的。

重新设置划分参数

当你点击了 **Default** 按钮，它将会变成 **Reset** 按钮。**Reset** 按钮允许你返回到大多数当前保存的设置（即点击 Default 之前所设置的值）。在执行之后，**Reset** 按钮将再变成 **Default** 按钮。

4.8.3 网格划分的方法

对于并行处理，划分网格有三个主要目的：

建立的划分有相等数目的单元

最小化划分界面的个数—即，降低划分边界表面的面积

最小化相邻划分的个数

平衡划分（调整单元的个数）保证了每个处理器有一个相等的负荷，并且划分将准备再同一时间通信。由于划分间的通信会是一个相对耗时间的程序，最小化界面的个数能够降低数据交换的时间。最小化相邻划分的个数减小了关于网络和线路争用的机会。再FLUNET中，划分方案使用对分运算法则建立划分，但不像其它方案，这些方案对划分的个数没有限制。对于每个可以的处理器，你将建立同一的划分个数（即，总的划分个数将是处理器个数的整数倍）。

对分方法

使用一个对分运算法则划分网格。所选的运算法则被用于父域，然后递归地被用于子域。例如，把网格分成四个划分，求解器将对分整个（父）域成两个子域，然后再分别对每个子域对分一次，于是得到了四个划分。要把网格分成三个划分，求解器将“对分”父域以建立两个划分—其中一个子域约是另一个的两倍，然后再把大的那个子域对分成两个，于是总共得到三个划分。

使用下面其中一个运算法则可以对网格进行划分。最有效的选择取决于问题本身，所以你可以尝试不同的方法直到你找到最适合你的问题的一个。

Cartesian Axes 基于单元的 Caresian 坐标对分域。它对分父域及所有随后产生的垂直于沿活动域最长范围的坐标方向的子域。

Cartesian Strip 使用坐标对分，但把所有的对分限制在父域最长范围的 Cartesian 方向。

Cartesian X-，Y-，Z-Coordinate 基于所选的 Cartesian 坐标对分域。它对分父域及所有随后产生的垂直于所规定的坐标方向的子域。

Cartesian R Axes 基于从单元中心到生成最小界面大小的 Caresian 轴（x，y，或 z）的最短径长而对分域。这个方法仅适用于 3D。

Cartesian RX-，RY-，RZ-Coordinate 基于从单元中心到所选的 Caresian 轴（x，y，或 z）的最短径长而对分域。这个方法仅适用于 3D。

Cylindrical Axes 基于单元的圆柱坐标对分域。这个方法仅适用于 3D。

Cylindrical R-，Theta-，Z-Coordinate 基于所选的圆柱坐标对分域。这个方法仅适用于 3D。

Metis 使用关于划分不规则图形的 **METIS** 软件包，它是由 Minnesota 大学和军事 HPC 研究中心的 Karypis 和 Kumar 发展而来的。它运用一个多级途径（在这里，细致图形上的顶点和边结合）形成一个粗糙的图形。粗糙的图形被划分，然后非粗糙的变回到最初的图形。在粗糙化和非粗糙化过程中，运算法则被应用来允许高质量的划分。

！注意，当使用插接版本（-pnet）时，METIS 划分不可用。在这种情况，METIS 划分可以使用划分过滤器获得。

Polat Axes 基于单元的极坐标对分域（图 4.8.9）。这种方法仅适用于 2D。

Polat R-Coordinat，Polar Theta-Coordinate 基于所选的极坐标对分域（图 4.8.9）。这种方法仅适用于 2D。

Principal Axes 基于一个与域的主轴结盟的坐标结构而对分域。当主轴与 Cartesian 轴结盟时，这减少了 **Cartesian** 对分（图 4.8.7）。这个运算法则也被称为瞬时、惯性、或惯性的瞬时划分。这个是 FLUENT 中缺省的对分方法。

Principal Strip 使用瞬时对分，但把所有的对分限制在父域最长范围的主轴上（图 4.8.8）。你可以经常使用这个方法最小化相邻划分的个数。

Princepal X-，Y-，Z-Coordinate 基于所选的主要坐标对分域（图 4.8.8）。

Spherical Axes 基于单元的球坐标对分域。这种方法仅适用于 3D。

Spherical Rho-，Theta-，Phi-Coordinate 基于所选的球坐标对分域。这种方法仅适用于 3D。

最优化

附加的最优化可以用来改进网格划分的质量。对于建立最小的界面边界，垂直于最长域范围的方向的划分不一定总是最好的选择。在划分之前，“预测验”操作可以用来自动地选择最好大方向。另外，存在如下的迭代最优化：

Smooth 试图通过交换划分之间的单元来最小化划分界面的个数。如果界面边界表面面积减小，这个方案横穿划分边界，且把单元送到邻近的划分。（如图 4.8.10 所示）

Merge 试图从每一划分消除孤生类簇。一个孤生类簇是有普通特性的一群单元，这个普通特性是群中的每个单元至少有一个与一界面边界符合的面。（图 4.8.11）孤生类簇能够使多网格执行退化，及导致大的通信消耗。

一般说来，Smooth 和 Merge 方案是相对便宜的优化工具。

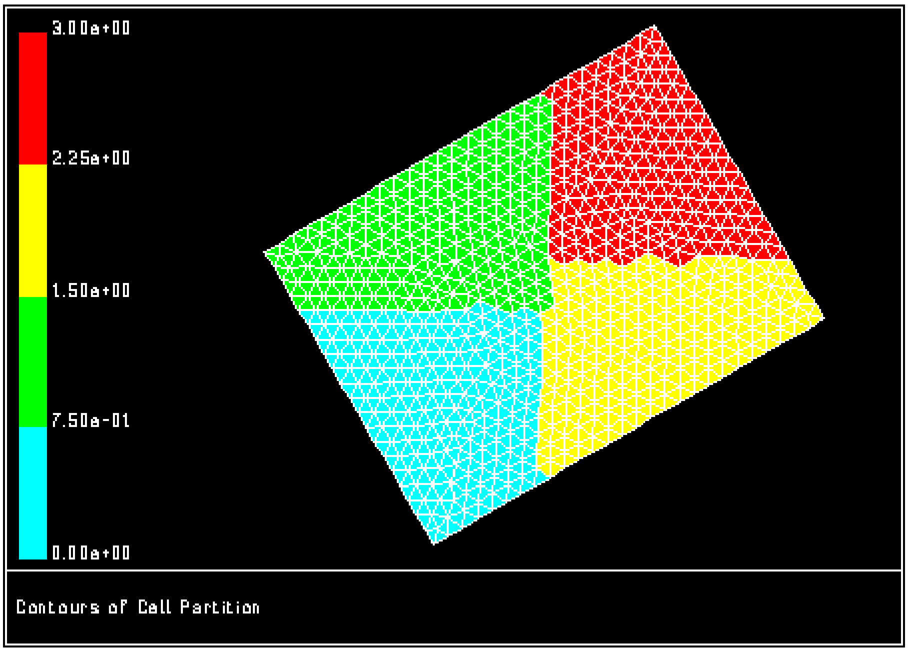


图 4.8.5：用 Cartesian Axes 方法建立的划分

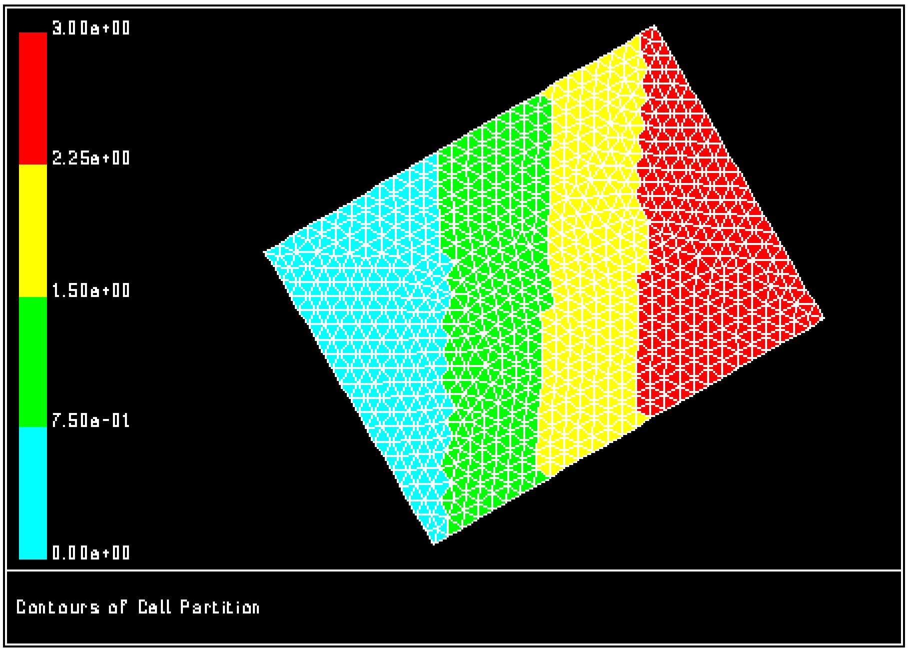


图 4.8.6：用 Cartesian Strip 或 Cartesian X-Coordinate 方法建立的划分

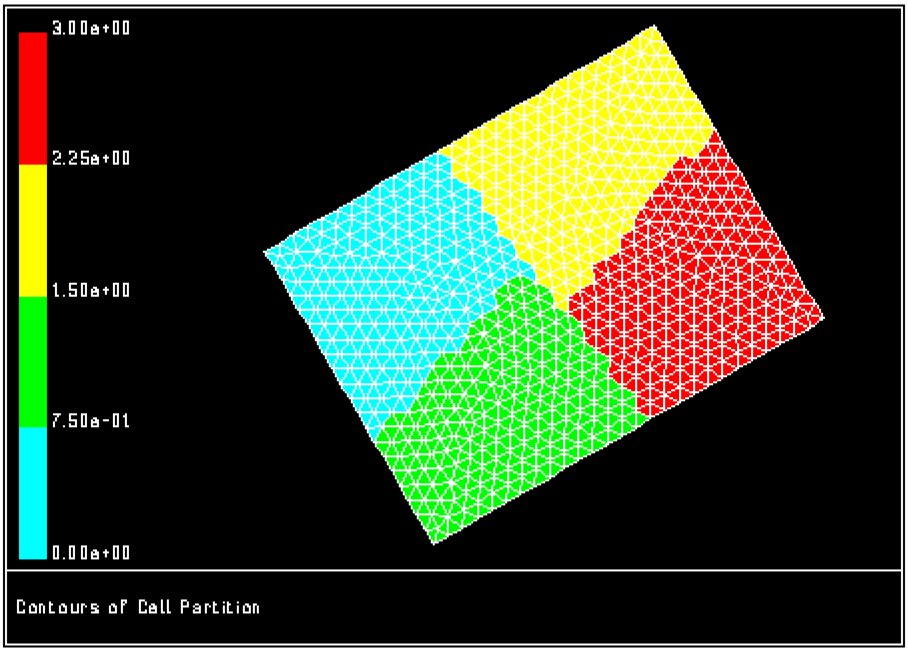


图 4.8.7：用 Principal Axes 方法建立的划分

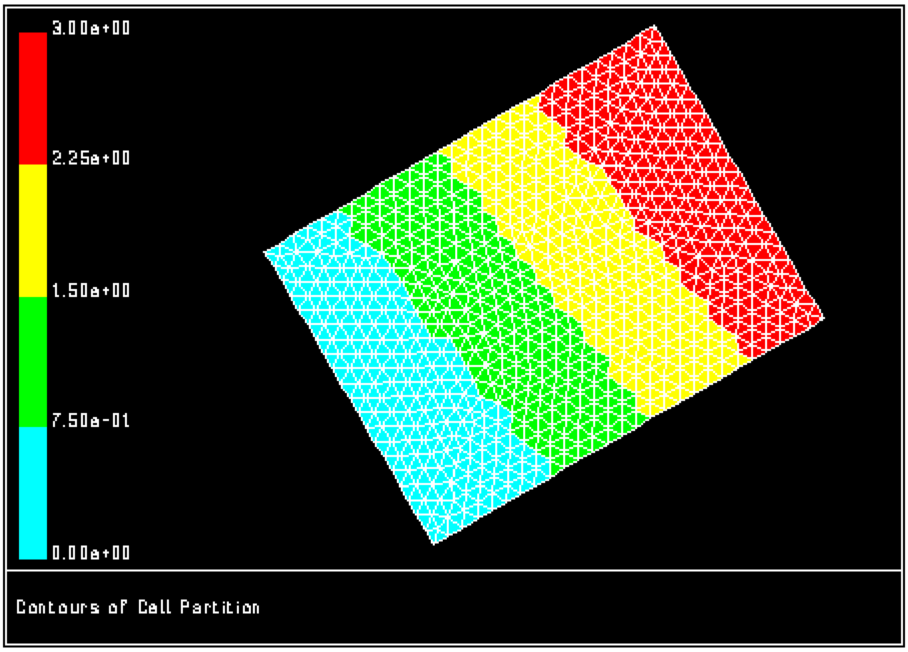


图 4.8.8：用 Principal Strip 或 Principal X-Coordinate 方法建立的划分

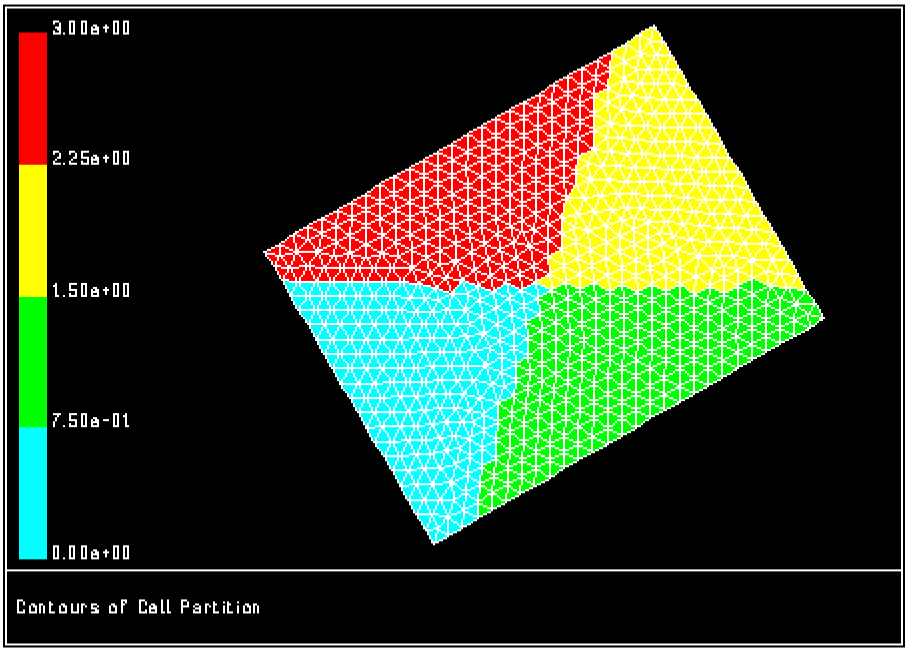


图 4.8.9：用 Polar Axes 或 Polar Theta-Coordinate 方法建立的划分

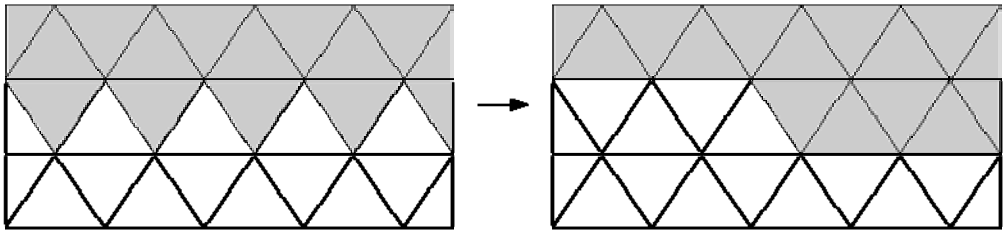


图 4.8.10：光滑最优化方案

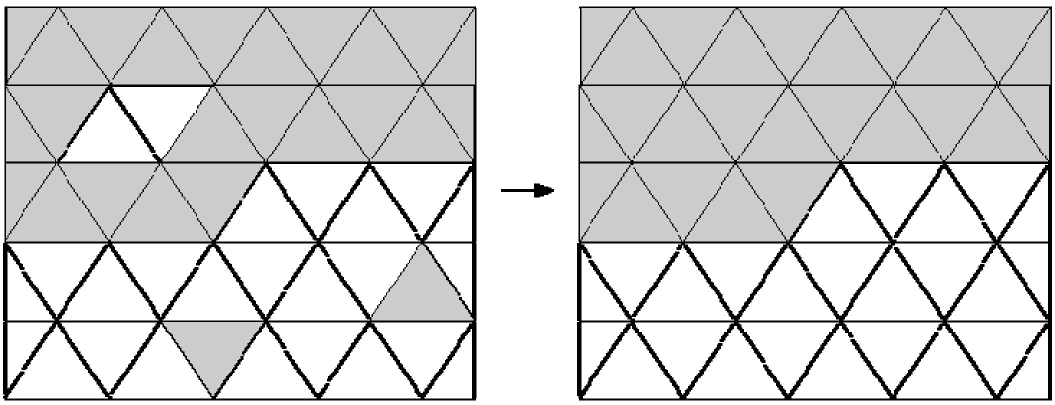


图 4.8.11：合并最优化方案

预测验

如果你选择 **Principal Axes** 或 **Cartesian Axes** 方法，在执行实际的对分之前，你可以通过测验不同指示以改进对分。如果你不选择使用预测验（缺省），那么 FLUENT 将执行垂直于最长域范围方向的对分。

如果预测验被激活，当点击 Partition Grid 面板中的 **Partition** 按钮时，或者当你正在使用自动划分读进网格时，它将自动地发生。对分运算法则将测验所有坐标指示及选择对最后对分产生最少划分界面地那个坐标指示。

注意，使用预测验将增加划分所需要的时间。对于 2D 问题，划分使花的时间是不用预测验的 3 倍，对于 3D 问题，它所花时间是不用预测验的 4 倍。

使用划分过滤器

除了在 Auto Partition Grid 和 Partition Grid 面板里之外，你也可以通过一个过滤器使用 METIS 划分方法。要对一个未划分的网格执行 METIS 划分，请使用

File/Import/Partition/Metes…菜单项。

File ⟶

Import ⟶
Partition ⟶
Metis…

FLUENT 将使用 METIS 划分器划分网格，然后把划分的网格读进求解器。划分的数目将等于处理器的数目。然后你可以处理模型定义和解。

！通过划分过滤器直接导入到并行求解器要求主机有足够的内存以运行所指定网格的过滤器。否则，你将需要在一台有足够内存的机器上运行过滤器。你可以在这台有足够内存的机器上运行并行求解器，并且重复前面所介绍的程序，或者在这台机器上手工地运行过滤器，然后把划分的网格读进关于主机的并行求解器。

要使用划分过滤器手工地划分一个网格，请键入下面的命令：

utility partition input-filename partition-count output-filename

这里，input-filename 是要被划分的网格的文件名，partition-count 是所要划分的个数，output-filename 是关于已划分的网格的文件名。然后，你可以把划分的网格读进求解器（使用标准的 File/Read/Case…菜单项）和处理模型定义和解。

当使用 File/Import/Partition/Metis…菜单项把一个未划分的网格导入到并行求解器时，METIS 划分器划分整个网格。你也可以使用 File/Import/Partition/Metis Zone…菜单项单独的划分每个单元域。

File ⟶
Metis Zone…
Partition ⟶

Import ⟶

这个方法对平衡工作负荷很有用。例如，如果一个 case 有一个流体域和固体域，流体域中的计算耗费要贵于固体域中的计算，所以单独的划分每一域将会更好的平衡工作负荷。

4.8.4　检查划分

划分一个网格之后，你应当检查网格信息及用图形检查划分。

解释划分的数据统计

你可以在划分（自动的和手工）完成之后你可以要求输出一个有关报告。在并行求解器中，点击面板 Partition Grid 中的 Print Active Partitions 或 Print Stored Partitions 按钮。在串行求解器中，点击 Print Partitions 按钮。

在一个并行问题中，FLUENT 区别两个单元划分方案：活动的单元划分和存储的单元划分。如果你使用 **Partition Grid** 面板先划分了网格，新的划分将被作为存储的单元划分。要使它成为活动的单元划分，你需要在 **Partition Grid** 面板中点击 **Use Stored Partition**s 按钮。活动的单元划分用于当前的计算中，而当你保存一个 case 文件时，用的则是存储的单元划分（最后被执行的划分）。由于有这两种单独的划分方案，你可以使用带有一定数目计算节点的并行求解器把一个网格细分成任意不同数目的划分，以适合于不同的并行机器，保存 case 文件，然后把它装载进指定的机器。

当你在串行求解器中点击 **Print Partitions** 时，你可以获得关于存储划分的信息。

你的目的是当保留一个平衡负荷（Cell variation）时，获得界面比率变化与全局界面比率的小值。

```
>>Partitions:
P    Cells  I-Cells  Cell Ratio  Faces  I-Faces  Face Ratio  Neighbors
0     134     10       0.075      217     10       0.046         1
1     137     19       0.139      222     19       0.086         2
2     134     19       0.142      218     19       0.087         2
3     137     10       0.073      223     10       0.045         1
   ------
  Partition count                =4
  Cell variation                 =(134 – 137)
  Mean cell variation            = ( -1.1% -    1.1%)
  Intercell variation            ＝  (10 – 19)
  Intercell ratio variation      = ( 7.3% -   14.2%)
  Global intercell ratio         = 10.7%
  Face variation                 = ( 4.5% -   8.7%)
  Global interface ratio         ＝  3.4%
  Nerghbor variation             ＝  ( 1 – 2)

Computing connected regions; type ^C to interrupt.
  Connected region count         =4
```

当一个 case 文件被读进并行求解器时，划分的 IDs 直接与计算节点的 IDs 一致。当一

个 case 文件中的划分个数大于计算节点的个数，但借助计算节点最终还是可分的，那么如此的分别已致 IDs 0 到（M - 1）的划分被映射到计算节点 0 上，IDs M 到（2M - 1）的划分被映射到计算节点 1 上，等等，这里 M 等于划分的个数与计算节点的个数之比。

用图形检查划分

为进一步解释划分的信息，你可以画出网格划分的轮廓，如图 4.8.5-4.8.9 所示。

Display→

Contours…

要展示活动的单元划分或存储的单元划分，请在 Contours Of 下面列里的 Cell Info…类中选择 Active Cell Partition 或 Stored Cell Partition，关掉 Node Values 的展示。

！如果你在建立你的问题过程中还没有如此做，那么为了使用 Contours 面板，你将需要执行一个解的初始化。

第五章　适应性网格技术

FLUENT 的解适应性网格细化的特性允许用户在几何的和数值的解数据的基础上细化和/或粗糙化网格。另外，为建立和查看用户化的适应场，FLUENT 提供了特殊应用的工具。本章从以下 10 节详细介绍适应性处理。

5.1 使用适应性网格

5.2 网格适应过程

5.3 边界适应

5.4 梯度适应

5.5 各向同性适应

5.6 区域适应

5.7 体积适应

5.8 y+和 y*适应

5.9 管理适应标识

5.10 适应性控制

5.11 用光滑和交换的方式改善网格

5.1　使用适应性网格

在 FLUENT 中，非结构网格的功能有两个重要的优势：

与结构网格相比，缩短设置时间

合并网格的解适应性细化的能力

使用解适应性细化可以添加网格中需要的单元，从而能够使流场得到更好的解决。当适应性被合适的利用时，由于解被用来决定添加更多单元的地方，因此所得到的网格对于流动的解是最理想的。换言之，网格中包含多余的单元不会浪费计算资源，这发生在结构网格中是比较有代表性的情形。进一步而言，网格细化对解的影响可以在不用完全重新生成网格的情况下来研究。

！在一个并行计算中，任何时候执行网格适应性，一个负荷平衡步将被 FLUENT 在缺省下执行。可以通过如下的命令关闭自动负荷平衡：

(disable-load-balance-after-adaption)

若返回到默认行为，则使用下面的命令：

(enable-load-balance-after-adaption)

注意，自动负荷平衡将不发生在与动态适应相关联的情况下。

5.1.1　适应性的例子

这一小节介绍怎样在可压缩、湍流流动经过一个 2D 涡轮叶栅的解中有效地使用适应性。最初的网格，如图 5.1.1 所示，在叶片周围是非常细的。表面节点分布为叶片几何提供了足够的定义，并且使湍流边界层不用进一步适应而能够得到完全解决。另一方面，进口、出口和周期边界处的网格是比较粗糙的。为保证在叶片通道里的流动得到适当地解决，解适应性细化被用来建立网格，如图 5.1.2 所示。注意，当例子涉及可压缩流动时，整个过程也适用于不可压缩流动。

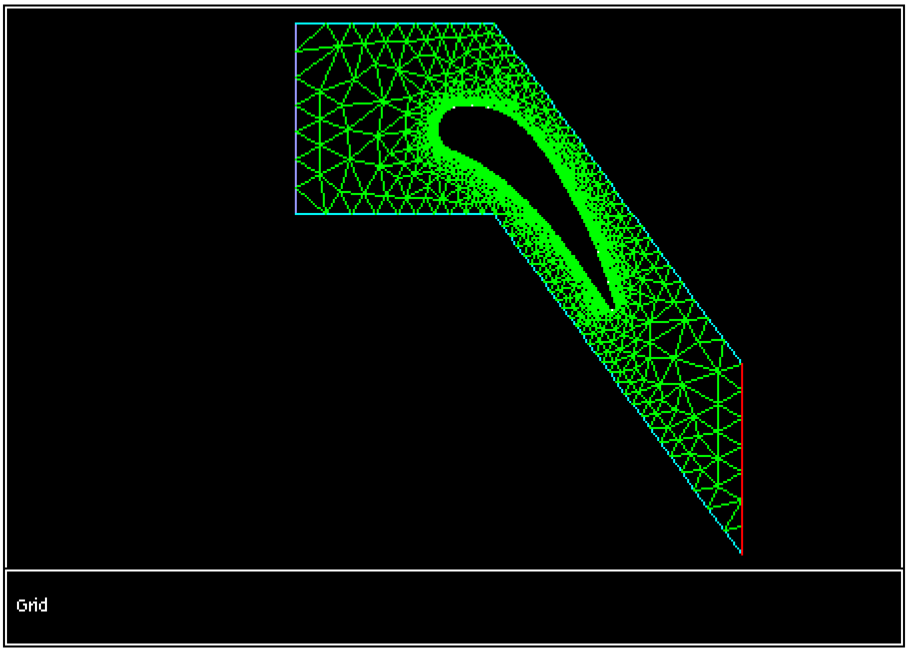


图 5.1.1：适应前的涡轮叶栅的网格

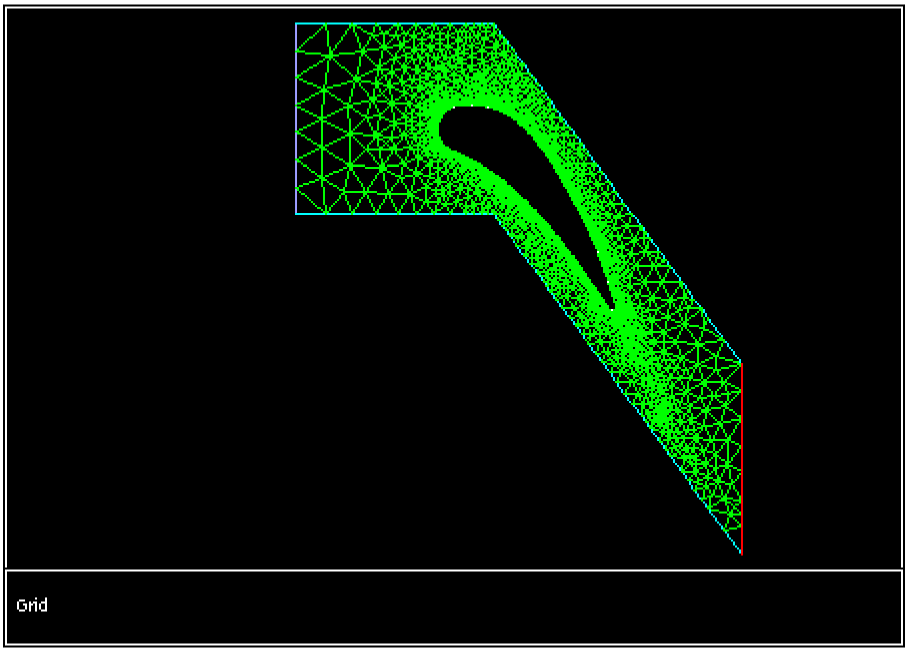


图 5.1.1：适应后的涡轮叶栅的网格

展示压力适应性函数的等值线已决定一个适合的细化临界。（参考第 5.4 节）

“标记”在细化临界内的单元，建立一个细化记录。（参阅第 5.2.1 节和第 5.4 节）

重复第 1 步和第 2 步的过程，使用作为一个细化标准的 Mack 数梯度。

要在尾流域中细分，请使用总压等值线作为一个标准。（参阅第 5.5 节）由于这些是高总压损失的区域，因此这造成了边界层内的单元和尾流要被做标记。

使用 Manage Adaption Registers 面板把三个细化记录合并成一个记录（参阅第 5.9 节）。

限制关于适应性的最小单元体积以阻止边界层内单元个数的增加（参阅第 5.10 节）。

细化包含在结果适应性记录中的单元。（参阅第 5.9 节）

使用 Smooth/Swap Grid 面板相继执行光滑和交换迭代。（参阅第 5.11 节）如果你正则使用正则适应性，建议执行这一步。

如图 5.1.2 所示，网格在叶片主边前面及叶片通道内是比较细的，在这里关于梯度上的细化效果很明显。在尾流区域内网格比较细是由于适应性使用了总压等值线。

5.1.2 适应的指导方针

从上一小节介绍的关于涡轮叶栅的例子可以直到，解适应细化的优势是显著的。然而，为避免一定的缺陷，解适应细化的功能必须谨慎运用。恰当运用它的一些指定方针如下：

表面网格必须细到能够表示几何的特性。

最初的网格应当包含充足的网格以捕获流场的重要特性。

在执行一个适应之前，一个收敛相当好的解应当被获得。如果你对一个错误的解实行

适应，那么单元会被添加到错误的流动区域。然而，因为在过早的对一个未收敛的解实行适应和当解没有太大的变化时继续迭代会浪费时间之间有个平衡问题，所以必须小心判断解的收敛情况。考虑到在每一迭代或每一时间步上（取决于所使用的求解器）对这儿的解做适应，所以不能直接应用于动态适应。

一般来说，在开始做适应性程序之前应当解一个 case 和 data 文件。那么，如果你生成了一个不想要的网格，你可以利用保存过的文件从新执行这个程序。和上一点的原因类似，这不适用于动态适应性。

当执行梯度适应时，你必须选择适当的变量。对于某些流动，选择是很清楚的。例如，关于细化冲击波区域，对压力梯度做适应是一个好的标准。然而，在大多数不可压缩流动中，对压力梯度做细化则意义不大。在一个不可压缩流动中，比较适合的参量大概要算平均速度梯度。

差的适应性实施会带来相反的效果。一个最一般的错误是过分细化一个特殊的解域，这会造成单元体积内的梯度过大。这样反而会影响解的准确性。

5.2 网格适应过程

适应过程被分成了两个明显的任务。第一个，单元个体基于适应方程被做了细化和/或粗化标记，它们是从几何和/或解数据建立的。下一步，单元基于这些适应标记被细化或被考虑粗化。这种模块化方法的主要优势在于不修改现存的网格而建立精细适应性函数和用各种适应性函数做实验的能力。

两种不同类型的适应在 FLUENT 中是可用的：“正则”和 “悬挂节点”适应。其中悬挂节点适应是缺省方法。而正则方法仅适用于三角形和四面体网格。

5.2.1 适应性和标记记录

激活 **Mark** 命令建立一个适应记录。一个适应记录基本上是一列关于域中每一个单元的标识符。标识符指定了一个单元是中性（被做标记的），还是细化的，还是粗化的。适应性函数是用来设置适当标识符的。例如，要基于压力梯度细化单元，求解器计算每个单元的梯度适应性函数。单元值被拿来与细化和粗化临界值做比较，然后被设置合适的标识符，特别对于下面的例子：

cell value <coarsen threshold: mark for coarsening

coarsen threshold<cell value<refine threshold: don’t mark, neutral

cell value>refine threshold: mark for refinement

GUI 和文本界面命令生成了适应记录。这些记录可以被转换成标记记录。Masks，不像适应记录，仅有两种状态：活动和不活动。如果适应记录被转换成一个标记，关于标记为细化的单元变成活动的，同时，那些未被标记的或标记为粗化的单元变成不活动的。你可

以使用一个标记记录来限制某个区域内的单元的适应。其过程图解说明如下。

图 5.2.1 给出了一些阴影单元，它们表示一个适应记录（阴影单元是被标记的单元）。图 5.2.2 演示了与一个标记记录相联系的活动单元。如果这个标记被应用于适应记录（即两者组合），那么合并所形成的新适应记录有标记的单元，如图 5.2.3 所示。关于合并记录更多的信息在第 5.10 节介绍。

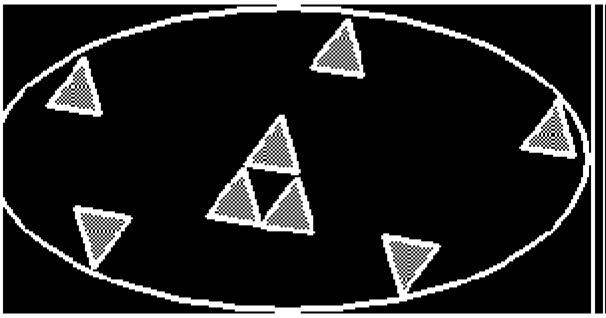


图 5.2.1：适应记录—阴影单元表示标记的单元

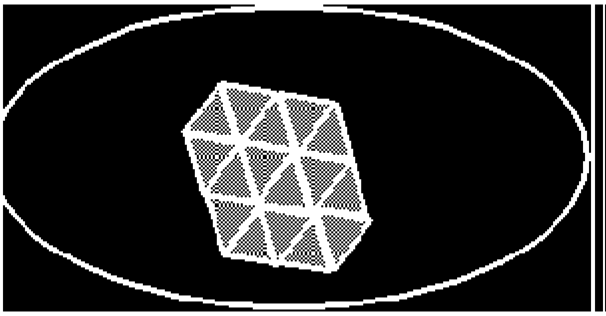


图 5.2.2：标记记录—阴影单元表示活动的单元

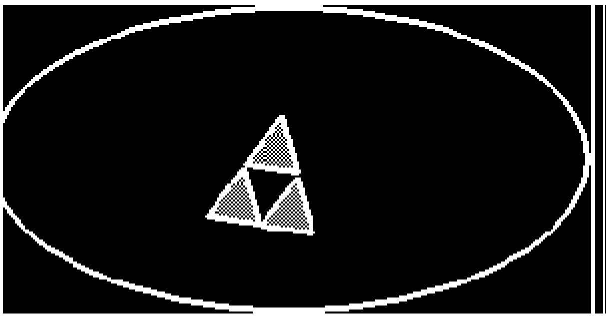


图 5.2.3：建立于记录应用的新适应记录

概括的讲，适应记录可以使用几何数据、流场的物理特性和它们的组合而建立。一旦建立，适应记录可以被列出、展示、删除、调换、改进、转化和改变成标记记录。

5.2.2 悬挂节点的适应

边和面上的节点，不是共享这些边或面的所有单元的顶点，这样的节点称为悬挂节点，如图 5.2.4 所示。悬挂节点网格的适应给出了操作各种类型的单元形状（包括混合网格）的网格。

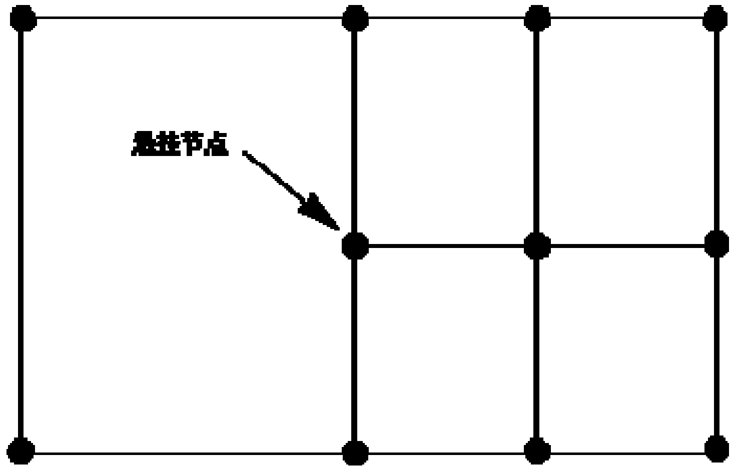


图 5.2.4：一个悬挂节点的例子

悬挂节点的细化

单元通过各向同性地再分每一个标记为细化的单元而被细化的。图 5.2.5 和 5.2.6 演示了所支持单元形状的分割，语言描述如下：

一个三角形被分成了 4 个三角形。

一个四边形被分成了 4 个四边形。

一个四面体被分成了 8 个四面体。

一个六面体被分成了 8 个六面体。

一个楔形（棱柱）被分成了 8 个楔形。

一个棱锥被分成了 6 个棱锥和 4 个四面体。

为保持精度，邻近单元的细化差别不允许超过一层。这阻止了适应生成过多差异的单元体积（降低了切断错误）和保证了“父”（最初的）和“子”（细化的）单元质心的相似性（降低了通量估值中的错误）。

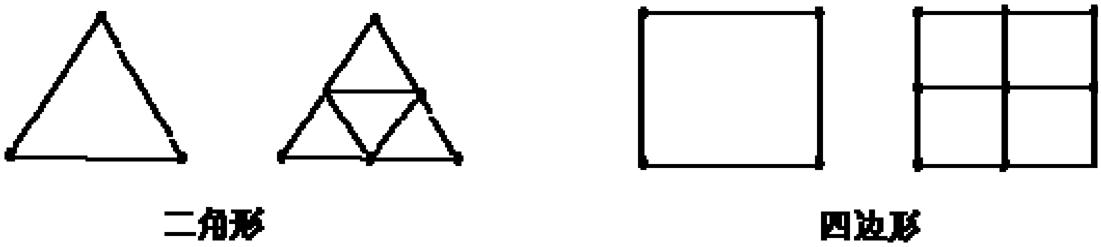


图 5.2.5：2D 单元类型的悬挂节点适应

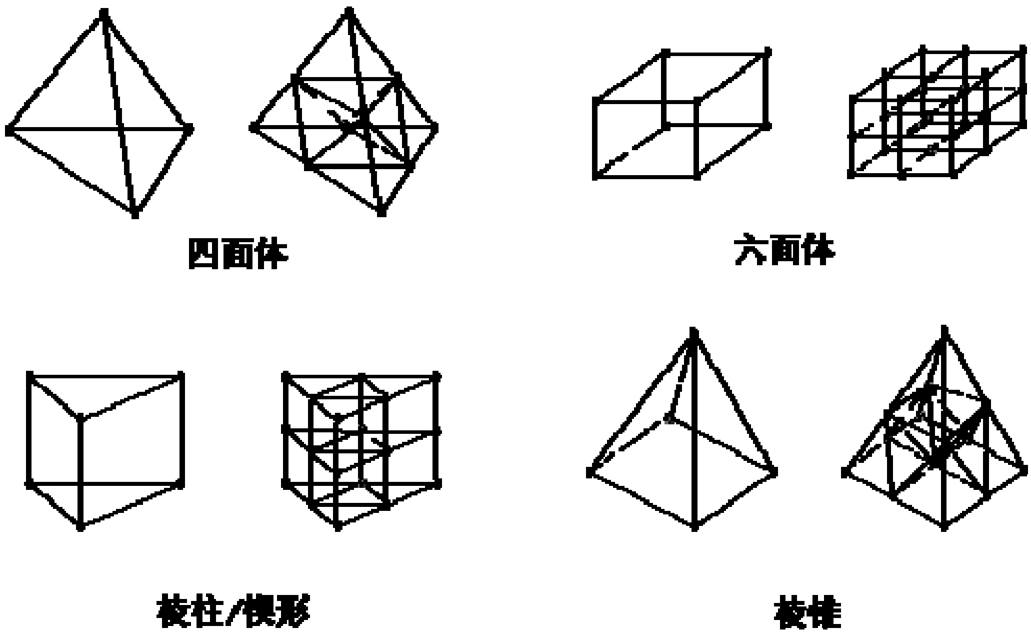


图 5.2.6：3D 单元类型的悬挂节点适应

悬挂节点的粗化

单元是通过再引入不活动的父单元被粗化的，即结合子单元以利用前面被细分的父单元。如果所有的子单元被标记为粗化，那么不活动的父单元被恢复活动。你最终将利用最初的网格，其中这个网格的悬挂节点粗化被重复应用。你不能使用悬挂节点适应方法比最初的网格更进一步的粗化网格。然而，粗化允许你移动最初的网格点以减少网格密度。

5.2.3　正则适应

正则适应过程不创建悬挂节点，而是共享一个边或面的所有单元都包含那些实体上的所有节点。注意，这个程序不适合与动态适应联合使用。

正则细化

要细化单元，你可以把边界或内部面（包括周期边界面）切开。图 5.2.7 给出了标签为 A 的三角形怎样被切开而细化的。单元通过分三角形或四面体的最长边被细分的。这一技术有两个好处：过程保守，不要求用插补来获得新单元的解矢量，另一方面，重复细分一个歪斜的单元不会继续增加网格的偏斜。

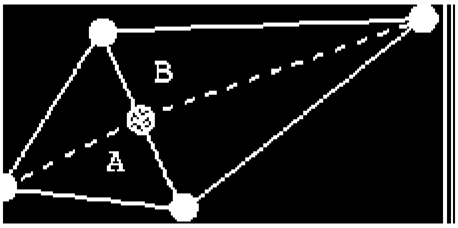


图 5.2.7：通过切分最长边而细化的单元

现在的这个方案假设已找到了标记为细化的任何单元的最长边。这个方案访问每个包含这样边的单元，并且寻找一个比较长的边。一旦它找到了最长的边，就会切开这条边。虽然这个程序保持了重复分成三角形的性质，但它也造成了许多没有标记为细化的单元被切开的现象。例如，图 5.2.8 给出了原来标记为细化的单元（标记为 X），图 5.2.9 给出了通过细化程序而建立的最终单元。

正则粗化

网格通过删除标记为粗化的单元上的节点而被粗化。如果依附在节点上的所有单元被标记为粗化，那么求解器企图删除节点。企图从标记为切除的每个节点出发而局部重分三角形的过程如下：

一列依附在标记节点上的单元被生成。切除这些单元将来一个必须被重新分成三角形的腔。

腔内的一列面被生成。

腔的边界上的一列面被生成。

如果要切除的节点在一个边界上，那么一个新边界的重分三角形被生成，及那些面被添加到腔上的这列面中。

从腔上的这列面出发，一个新的 Delaunay 重分三角形被建立。

如果程序成功，区域中原始三角形上的节点、面和单元被删除。

所有与腔相关的节点被从这列命定的节点中切除以避免在同一区域中产生连续的粗化。

在新单元中的解变量是使用一个体积的加权平均计算的。

图 5.2.10 给出了节点 n1 的切除和区域被重新分成三角形的结果。在这个例子中，附在节点上的这列单元包括 c1，c2，c3，c4 和 c5；腔内的这列面包括 f6，f7，f8，f9 和 f10；

腔上的这列面包括 f1，f2，f3，f4 和 f5。重分三角形的新面是 f11 和 f12，新单元是 c6，c7 和 c8。

由细化所引入的节点被称为细化节点。网格中的节点在细化之前被称为最初（或原始）节点。缺省时，仅有细化节点可以在粗化过程中被切除，但是你可以通过重新设置节点标志而切除任何节点。关于节点标志的内容在第 5.10 节中有详细介绍。

目前，网格粗化的方案只可以在 FLUENT 的 2D 版本中执行。

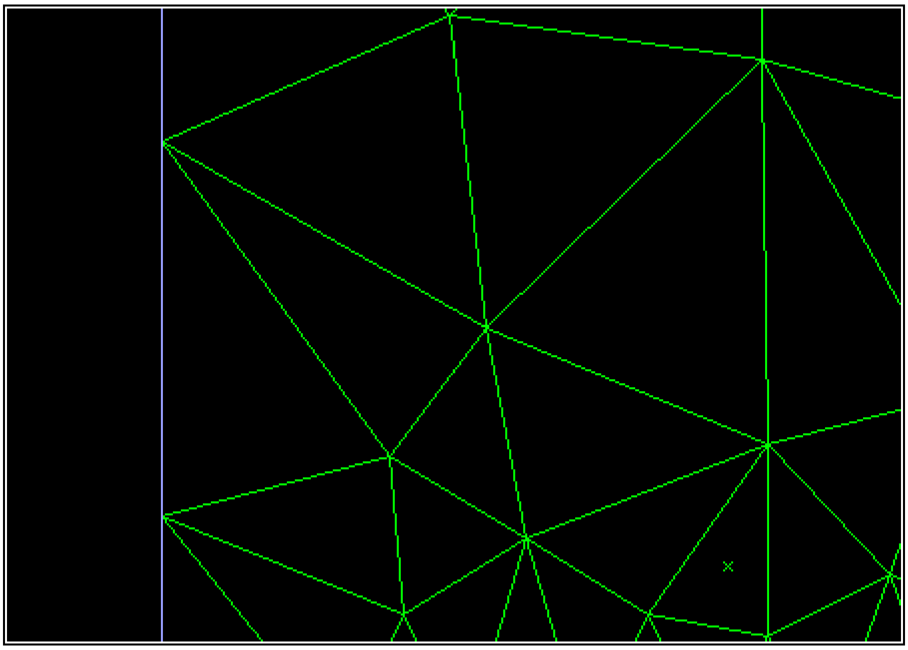


图 5.2.8：包含一个标记为细化的单元的原始网格

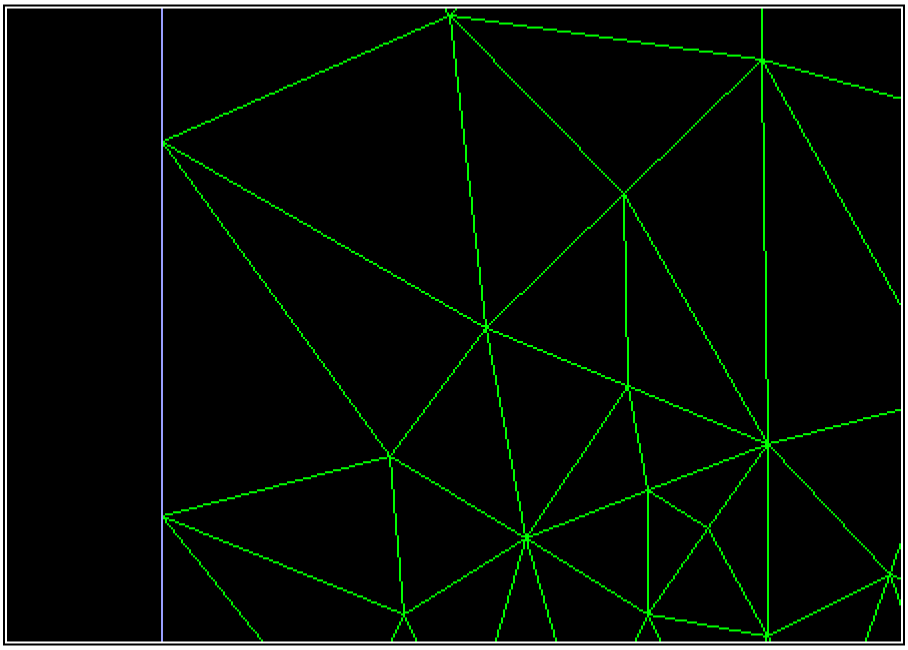


图 5.2.9：细化程序之后的最终网格

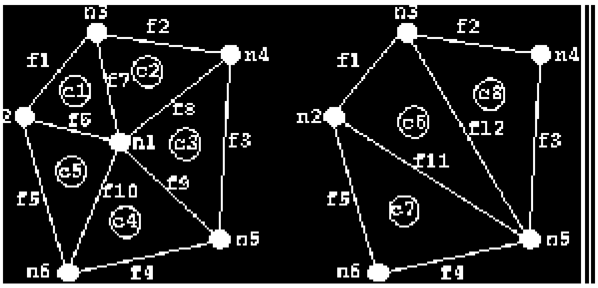


图 5.2.10：通过切除一个节点及对域重分三角形而粗化网格

5.2.4 正则与悬挂节点的适应性对比

对于大多数问题，悬挂节点的适应性为网格适应提供了相当大的机动性。然而，下面几点将会帮助你针对特殊的应用选择适当类型的适应性。

正则适应性的方法仅适用于三角形和四面体网格，而悬挂节点适应性则可以应用于所有被支持的单元类型。

悬挂节点适应性通常在本质上比正则适应性更具有局部性。在正则适应中，除了被标记的单元，许多单元可以由最长的边分开标准而被细化。对于高级别的网格，原始的正则细化延伸趋于展示单元细化的真实传播，有时细分许多远离标记为细化的实际单元的网格单元。悬挂节点方案仅宣传维持细化水平的差异，所以比较有局限性。

原始网格的连通性在悬挂节点适应性方案中仍保持，但正则适应性方法将修改关于细

化和粗化适应的连通性。对于网格被用于一个有周期行为的非稳定问题（例如，一个圆柱体后面的涡脱落），如果你执行连续的细化和粗化，这可能牵连着精确性问题。然而，仅有正则粗化允许你粗化原始的网格，并且这仅适用于 2D 问题。

悬挂节点适应性有一个与维持网格层次和临时存储 3D 中的边相联系的内存惩罚问题。正则适应性除了附加的节点、面和单元被添加以增加网格密度外则没有内存费用问题。

正则适应性不应当用于动态适应性。

5.3　边界适应

如果一个边界上需要更多的单元，那么可以使用边界适应来添加单元。边界适应函数允许你在邻近所选择的边界域内对单元进行标记或细化。之所以 FLUENT 有对邻近一个或多个边界域的单元进行细化的能力，是因为重要的流体交互作用经常发生在这些区域，比如强速度梯度在接近一个壁面的边界层内发展。

5.3.1　边界适应的例子

图 5.3.1 给出了一个关于网格可以使用边界适应而改进的例子。这个网格在台阶的竖直面上仅有两个单元。在与这个竖直面相应的区域上，边界适应可以被用来增加单元的数目，如图 5.3.2 所示。注意，这个步骤不被增加一个弯曲表面的分解。所以，如果在一个比较重要形状的曲面上需要更多的单元，最好在把网格读进求解器之前建立这个含有重要表面节点的网格。

5.3.2　执行边界适应的步骤

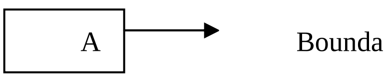
三种不同的方法可以用于边界适应：

基于一个单元到边界的距离的适应，在数单元中被测量

基于一个单元到边界的正则距离的适应

基于一个目标边界体积和增长因子的适应

你可以在 **Boundary Adaption** 面板（如图 5.3.3 所示）中使用这三种方法的任一个。



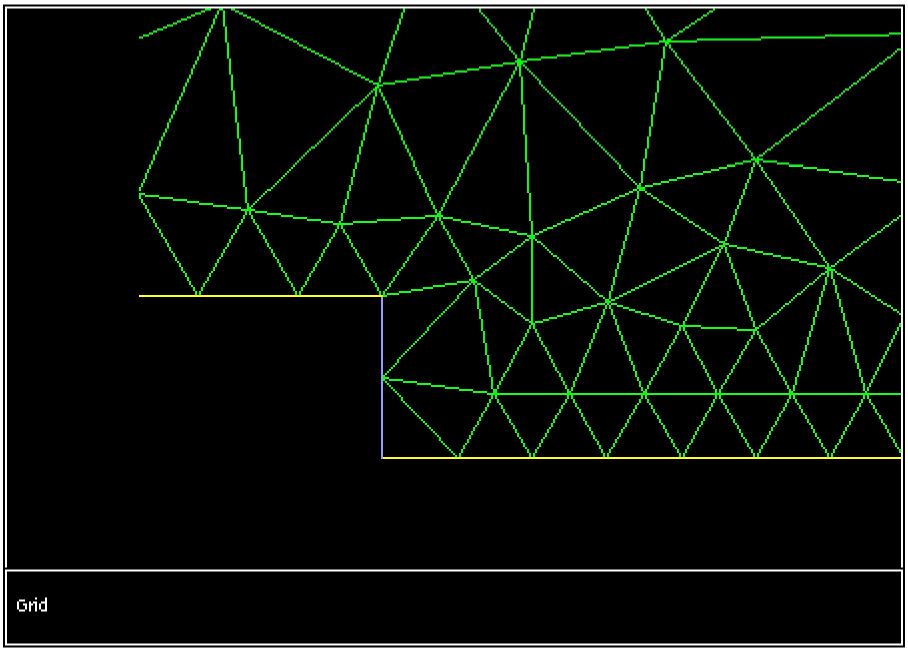


图 5.3.1：边界适应前的网格

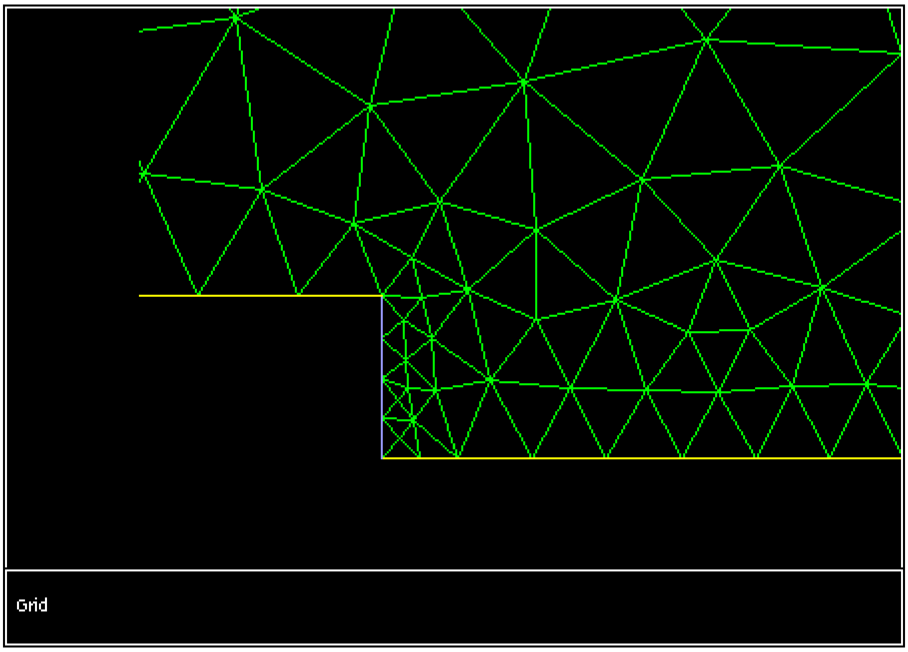


图 5.3.2：边界适应后的网格

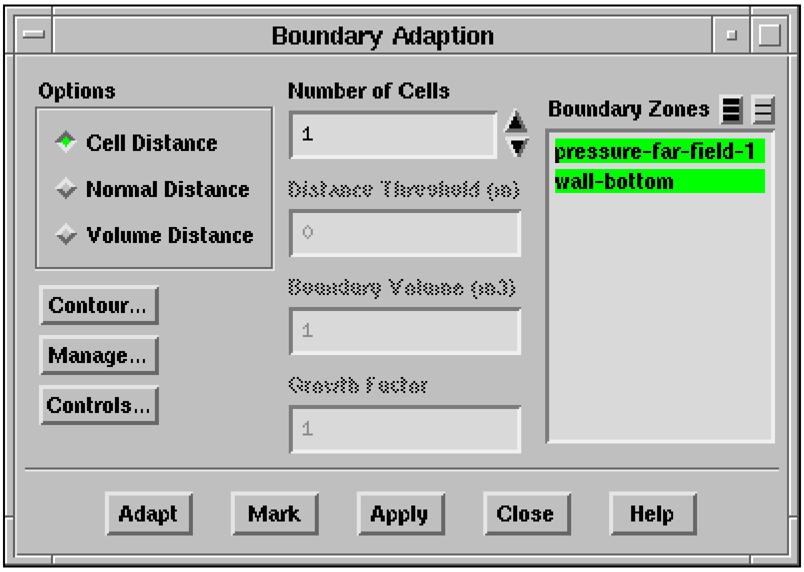


图 5.3.3：Boundary Adaption 面板

基于单元个数的边界适应

从单元的个数出发，在一个单元到边界的距离的基础上执行适应的一般步骤如下：

1. 在 Boundary Adaption 面板上，选择 Options 下面的 Cell Distance，在 Boundary Zones 列中选择你想细化的单元所邻近的边界域，然后点击 Apply。

2. 点击 Contour…按钮打开 Coutours 面板。

3. 在 **Contours** 面板中，激活 **Filled**，使 **Node Values** 不活动，在 **Contours Of** 下面的列中选择 **Adaption…**和 **Boundary Cell Distance**，选择合适的表面（仅对 3D），点击 **Display** 查看单元的位置，其中每个单元都有边界单元距离值。通过展示值的不同范围，你可以决定你想进行适应的单元的单元距离。

4. 在 **Boundary Adaption** 面板中，设置 **Number of Cells**。如果你保留缺省值 1，那么仅有那些在指定的边界域上有边（2D）或面（3D）的单元（即一个边界单元距离为 1 的那些单元）将被标记或适应。如果把值增加到 2，一个边界单元距离为 2 的单元也将被标记/适应，依此类推。

5.（可选择的）如果你想设置任何适应选项（在第 5.10 节有介绍），点击 **Controls…** 按钮打开 **Grid Adaption Controls** 面板。

6. 点击 **Mark**，这样通过把单元放进一个适应记录而标记单元为细化，或着点击 **Adapt** 立即执行细化。

基于正则距离的边界适应

关于执行基于一个单元到边界的正则距离的细化的一般步骤如下：

1. 在 Boundary Adaption 面板，选择 Options 下面的 Normal Distance，在 Boundary Zones

列中选择你想细化的单元所邻近的边界域，然后点击 Apply。

2．点击 Contour…按钮打开 Contours 面板。

3．在 **Contours** 面板中，激活 **Filled**，使 **Node Values** 不活动，在 **Contours Of** 下面的列中选择 **Adaption…**和 **Boundary Normal Distance**，选择合适的表面（仅对 3D），然后点击 **Display** 查看单元的位置，其中每个单元都有正则距离值。通过展示值的不同范围，你可以决定你想进行适应的单元的正则距离。

4．在 **Boundary Adaption** 面板中，设置你所需要 **Distance Threshold** 值。到所选择的边界区域的一个正则距离小于或等于这个值的单元将被标记或适应。

5．（可选择的）如果你想设置任何适应选项（在第 5.10 节有介绍），点击 **Controls…**按钮打开 **Grid Adaption Controls** 面板。

6．点击 **Mark**，这样通过把单元放进一个适应记录而标记单元为细化，或着点击 **Adapt** 立即执行细化。

基于目标边界体积的边界适应

这个边界适应允许你在一个目标边界单元体积和指数增长函数的基础上细分单元。单元基于下面的方程被标记为细化：

$$V_n > V_{boundary} e^{\alpha d} \quad (5.3.1)$$

其中，V_n 是单元的体积，$V_{boundaary}$ 是规定的边界体积（Boundary Volume），α 是指示增长因子（Growth Factor），d 是单元质心到所选边界的正则距离。$V_{boundary} e^{\alpha d}$ 是一个单元的目标体积。

关于这种类型的边界细化的一般步骤如下：

1．在 Boundary Adaption 面板，选择 Options 下面的 Volume Distance，设置 Boundary Valume 和 Growth Factor 的值，在 Boundary Zones 列中选择你想使用的边界区域，然后点击 Apply。

2．点击 Contour…按钮打开 Contours 面板。

3．在 Contours 面板中，激活 Filled，使 Node Values 不活动，在 Contours Of 下面的列中选择 Adaption…和 Boundary Normal Distance，选择合适的表面（仅对 3D），然后点击 Display 查看目标体积的等值线。你可以修改任何输入的值（Boundary Volume，Growth Factor，和 Boundary Zones），在 Boundary Adaption 面板中点击 Apply，然后再展示等值线图以显现修改的目标体积分布。

4．（可选择的）如果你想设置任何适应选项（在第 5.10 节有介绍），点击 **Controls…**按钮打开 **Grid Adaption Controls** 面板。

5．点击 **Mark**，这样通过把单元放进一个适应记录而标记单元为细化，或着点击 **Adapt** 立即执行细化。

5.4 梯度适应

梯度适应函数允许你在所选择的场变量的梯度（或曲率）基础上标记单元或适应网格。

5.4.1 梯度适应途径

解适应的网格细化的基本目的是有效地降低数值解中的数值错误，最小化数值成本。可惜的是，由于在适应网格中正确地估计和对误差建模都比较复杂，因此关于点插入适应方案的直接误差估计是困难的。事实上，关于误差估计和收敛，在数学上还没有全面严格的理论可以用于 CFD 模拟。假设最大的误差发生在高梯度区域，展开流场是容易用的物理特性，它可以用来推进网格适应的进程。对网格适应来说，使用这个信息在 FLUENT 中有两个途径：

第一个梯度途径建议在强冲击的问题中使用，例如，超音速无粘流动。在这个途径中，一个特征长度乘上所选解变量的梯度的未划分的 Euclidean 标准。例如，两维中的梯度函数的形式如下：

$$|e_{i1}| = (A_{cell})^{\frac{r}{2}} |\nabla f| \quad （5.4.1）$$

这里，e_{i1} 是误差指标，A_{cell} 是单元的面积，r 梯度体积重量，∇f 是场变量 f 的梯度的未分割的 Euclidean 标准。梯度体积重量的缺省值是统一的，它符合满体积加权；一个 0 值将消除体积加权，0 到 1 之间的值将使用体积的比例加权。

第二个梯度（曲率）途径建议用于流畅的解问题。这个是 FLUENT 原来使用的等分布适应技术，一个特征长度乘上所选解变量的未分割的 Laplacian 算子。例如，二维中梯度函数的形式如下：

$$|e_{i2}| = (A_{cell})^{\frac{r}{2}} |\nabla^2 f| \quad （5.4.2）$$

其中，e_{i2} 是误差指标，A_{cell} 是单元的面积，r 梯度体积重量，$\nabla^2 f$ 是场变量 f 的未分割的 Laplacian 算子。梯度体积重量的缺省值是统一的，它符合满体积加权；一个 0 值将消除体积加权，0 到 1 之间的值将使用体积的比例加权。

长度是单元体积的平方根（2D）或立方根（3D）。长度的引入均允许强和弱干扰的分解，从而为获得更精确的解增加潜力。然而，你可以通过在 **Grid Adaption Controls** 面板中改变梯度 **Volume Weight** 而减少或消除体积加权。

任何适用于做等值线的场变量都可以用于梯度适应函数。有趣的是，这些刻度函数包括数值解的几何和物理两种特性。所以，除了对物体特性的传统适应，例如速度，你可以选择对单元体积场进行适应以减少单元体积内的快速变化。

除了 FLUENT 原来用的 **Standard**（非规格化）途径，下面两个选项适应于 **Normalization**：

1．**Scale**，它是 e_{i1} 或 e_{i2} 的值与其在区域内的平均值的比例，即：

$$\frac{|e_i|}{|\bar{e}_i|} \quad (5.4.3)$$

当使用 **Scale** 选项，关于 **Coarsen Threshold** 和 **Refine Threshold** 的适当的首切值分别是 0.3 到 0.5，和 0.7 到 0.9。较小的值将导致较大的适应的区域。

2．**Normalize**，它是 e_{i1} 或 e_{i2} 的值与其在区域内的最大值的比例，即：

$$\frac{|e_i|}{\max|e_i|} \quad (5.4.4)$$

当使用 **Normalize** 选项时，关于 **Coarsen Threshold** 和 **Refine Threshold** 的适当的首切值分别是 0.2 到 0.4，和 0.5 到 0.9。较小的值将导致较大的适应的区域。

5.4.2　稳定的梯度适应例子

越过一个循环圆柱体的超音速流动的解是使用梯度适应的一个比较好的例子。初始网格，如图 5.4.1 所示，即使包含了足够能描述圆柱体形状的网格，但仍很粗糙。圆柱体前头的网格太粗糙以致不能解决形成于圆柱体前面的冲击波。在这个例子中，在经过冲击的压力中将有一个跳跃，所有显然压力适合用于梯度适应。然而，在冲击能够被适当解决之前，必须有几个适应。几个适应之后的网格如图 5.4.2 所示。

关于一个不可压缩流动的梯度适应的一个有代表的应用可能是一个涉及一中断的混合层。

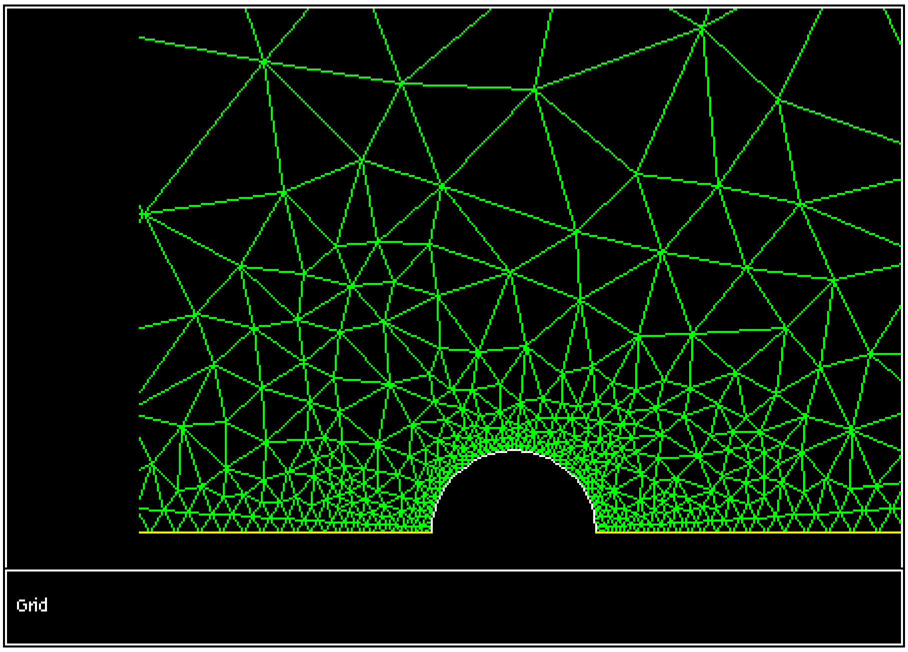


图 5.4.1：适应之前的非流线型体网格

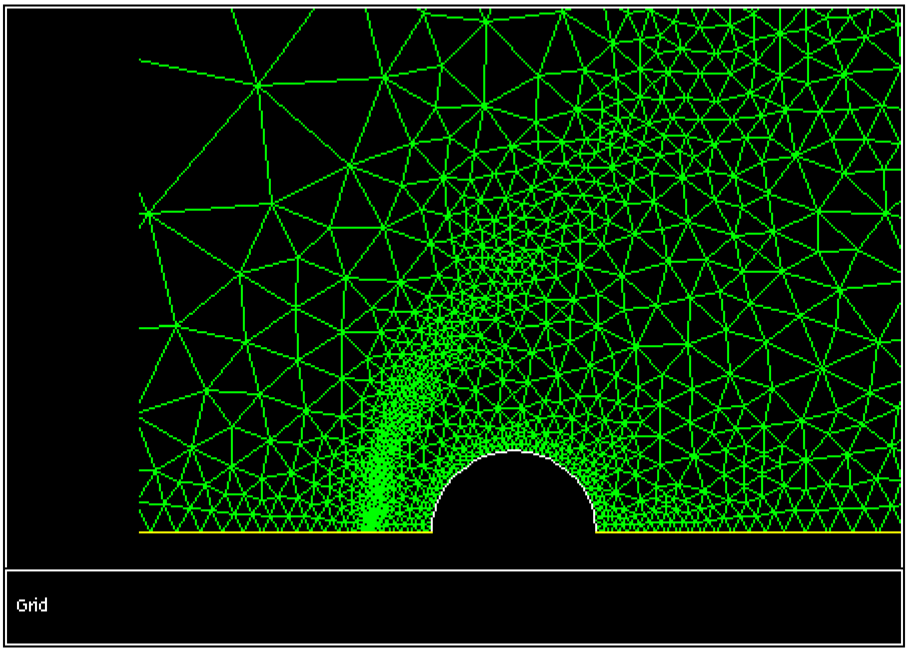


图 5.4.2：梯度适应之后的非流线型体网格

5.4.3 执行梯度适应的步骤

在 **Gradient Adaption** 面板（如图 5.4.3 所示）中执行梯度适应。

A → Gradien

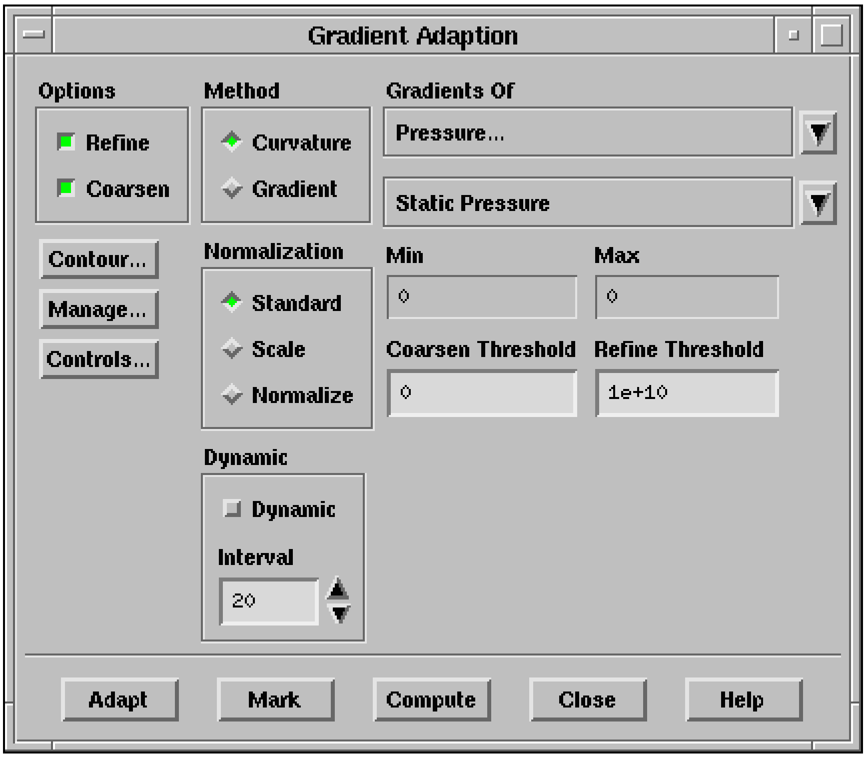


图 5.4.3：Gradient　Adaption 面板

执行梯度适应的一般步骤如下：

1．在 Grandient Adaption 面板中，选择 Gradients Of 下面的列中所需要的解变量，然后点击 Compute。

2．点击 Contour…按钮打开 Contours 面板。

3．在 **Contours** 面板中，激活 **Filled** 等值线，使 **Node Values** 不活动，在 **Contours Of** 下面的列中选择 **Adaption…**和 **Existing Value**，选择合适的表面(仅对 3D)，然后点击 **Display** 查看单元的位置，其中每个单元都有梯度值。通过展示值的不同范围，你可以决定你想使适应的单元的梯度范围。注意，如果你使用规格化，那么任何变量的梯度的范围将保持在[0, 1]。

4．在 **Gradient Adaption** 面板中设置 **Refine Threshold**。有梯度值的单元将被标记或细分。在 **Gradient Adaption** 面板中，你也会发现类 **Method** 和 **Normalization**。Curvature 是关于 FLUENT 原来所使用的适应的标准；Grandient 是一个比较新的标准，为了适应性，它使用一个场变量的第一个梯度，而不是第二个梯度（曲率）。取决于所求解的问题，你可以使用一个或其它的适应方法：

Curvature 被建议用于流畅解的问题。

Gradient 被建议用于强冲击的问题，例如，超音速无粘流动。

进一步的选项还有：

Standard 意味着没有规格化的梯度或曲率被执行。这中情形一直存在于 Fluent 6.0 中。

Scale 意味着梯度或曲率是对区域中的平均值的比例。

Normalize 使用一个对域中变量的最大值的一个缩放比例，即梯度或曲率的取值范围为[0, 1]。

使用缩放比例或规格化使细化和粗化临界值的设置变地简单了，并且几乎独立于当前的解和特定问题。这对于使用自动的动态程序尤其重要。

5．如果你想粗化网格，设置 Coarsen Threwshold 为非零值。带有梯度值（在特定值下面）的单元将被标记或粗化。

注意，如果你使用的是悬挂节点适应（缺省），那么你将不能建立一个比原来更粗的网格。对于这种情况，你必须使用正则适应。还需要注意的是，正则粗化仅适用于 2D 或轴对称几何问题。详细内容请看第 5.11 节。

6．如果你想设置任何适应选项（在第 5.10 节有介绍），点击 **Controls…**按钮打开 **Grid Adaption Controls** 面板。

7．点击 **Mark**，这样通过把单元放进一个适应记录而为适应（细化/粗化）给单元做标记，或着点击 **Adapt** 立即执行适应。

（如果你希望使细化或粗化不活动，或者为细化或粗化做标记，那么你可以在做标记或适应之前关掉 **Refine** 或 **Coarsen** 选项。）

5.5 各向同性适应

一些流动可能包含一些流动特性，这些特性基于一定量的值是容易区别的。例如，波表示一个总压不足，喷射可以通过相对高速流的一个区域而确定。由于这些区域也包括一些重要流动量（例如湍流中的 k 和 ε ）的大梯度，因此在执行一个关于相关流动量的各向同性适应可能比细化个别流动变量的梯度要方便些。

各向同性适应函数允许你标记或细分单元，这些单元位于一个对挑选的场变量函数所规定的范围的内部或外部。网格可以基于几何或解矢量数据被细化或为细化而被标记。特别是，关于场变量的显示列中的任何量都可以被由于各向同性适应。下面介绍了一些关于怎样使用各向同性标记/适应的特性的例子：

使用坐标值或二次曲面函数建立掩饰。

细分单元，其中单元有一个不超出规定范围的速度值。

单元有一个压力或连续残差在一个所要求的范围之外，标记和展示这样的单元以确定数值解快速变化的位置。

用于各向同性适应的函数的途径是为每个单元计算所指定的值（速度，二次曲面函数，质心的 x 坐标，等等），然后访问每个单元，为细化对单元做标记，其中这些单元有在规定范围内（或外）的值。

5.5.1　各向同性适应的例子

各向同性在一个问题中有用的例子如图 5.5.1 所示。关于一个撞击射流的网格与 x 速度等值线一起在图中展示。基于 x 速度的各向同性适应仅允许在射流中的网格细化，细化结果如图 5.5.2 所示。

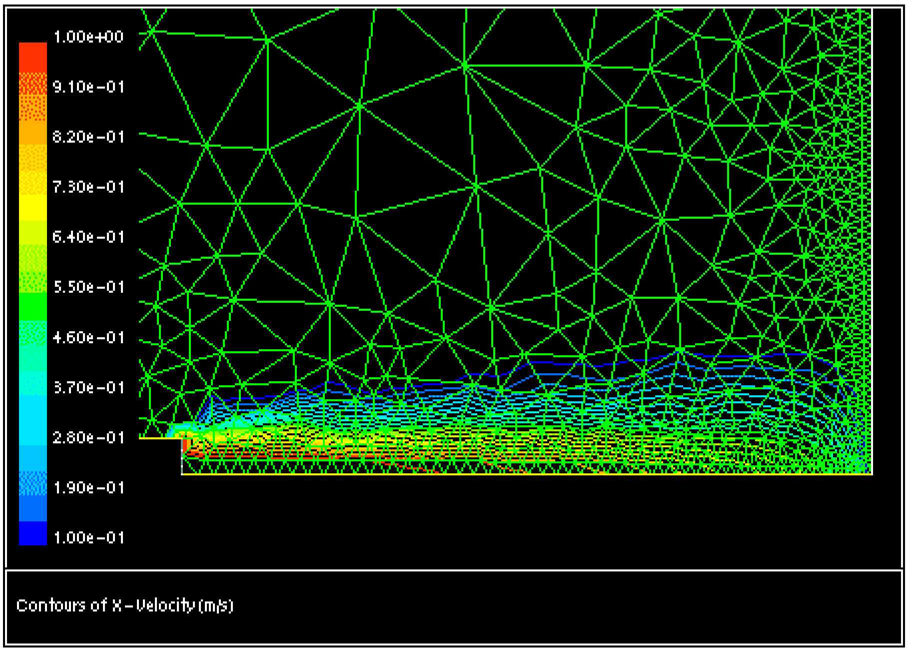


图 5.5.1：在适应前的撞击射流网格和 x 速度等值线

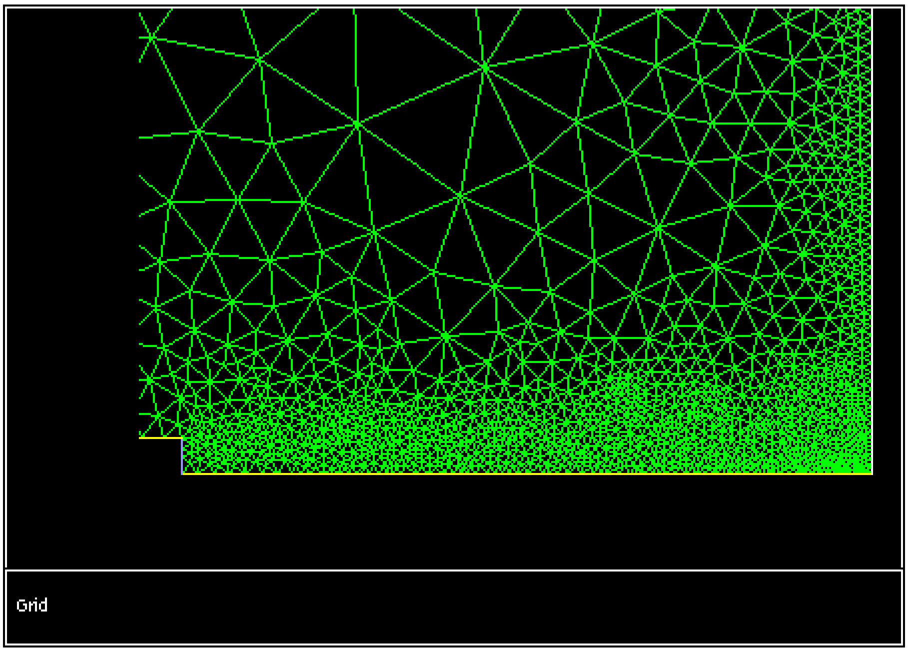


图 5.5.2：各向同性适应之后的撞击射流网格

5.5.2 执行各向同性适应的步骤

在 **Iso-Value Adaption** 面板（如图 5.5.3 所示）中执行各向同性适应。

A

Iso-Valu

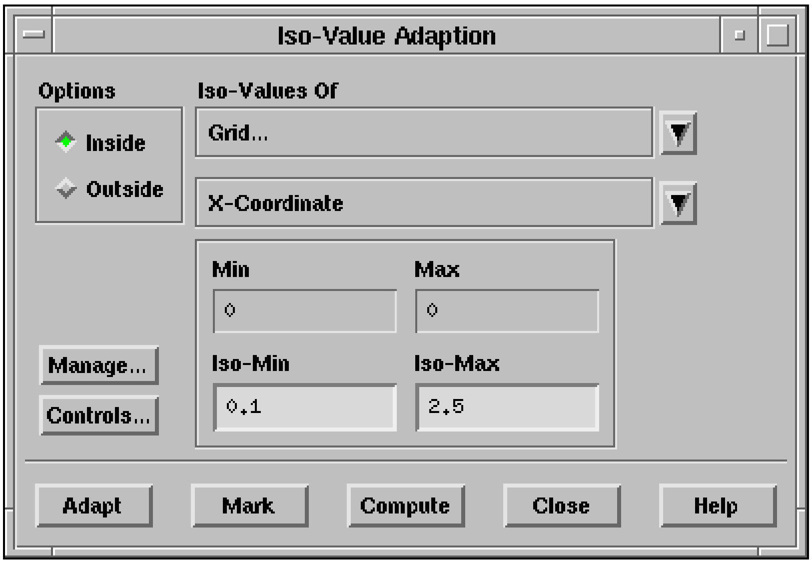


图 5.5.3：Iso-Value　Adaption 面板

执行各向同性适应的一般步骤如下：

1．在 Iso-Value Adaption 面板中，选择 Iso-Values Of 下面的列中所需要的解变量，然后点击 Compute 去更新 Min 和 Max 场。

2．选择 Inside 或 Outside 选项，并设置 Iso-Min 和 Iso-Max 的值。

如果选择 **Inside**，在 **Iso-Min** 和 **Iso-Max** 值之间的各向同性值的单元将被标记或细分。

如果选择 **Outside**，低于 **Iso-Min** 和高于 **Iso-Max** 的各向同性值的单元将被标记或细分。

3．（可选择的）如果你想设置任何适应选项（在第 5.10 节有介绍），点击 **Controls…** 按钮打开 **Grid Adaption Controls** 面板。

4．点击 **Mark**，这样通过把单元放进一个适应记录而为细化给单元做标记，或着点击 **Adapt** 立即执行适应。

5.6　区域适应

如果解域的某个区域需要更细的网格，那么可以使用区域适应来细化网格。区域适应函数标记或细分一个区域内部或外部的单元，其中这个区域是由文本或鼠标输入而定义的。目前，网格可以在一个六面体（2D 中是四边形）、球体（在 2D 中是圆）或圆柱体的内部或外部被细分或标记。基于标记/适应特性的区域对细分区域特别有用。另外，你可以使用区域标记来建立掩饰适应记录，其中掩饰适应记录可以用来限制细化和粗化的范围。

5.6.1　定义一个区域

区域适应函数的基本方法是先定义一个六面体的（四边形的）、球形的（圆的）或圆柱的区域。通过输入定义对角线的两个点的坐标来定义六面体（四边形）。通过输入球心的坐标和半径来定义球（圆）。要定义一个圆柱（cylinder），只要指定定义圆柱轴的点的坐标及半径的长度即可。在 3D 中，这将定义一个圆柱体；在 2D 中，这将是一个任意定位的矩形，其长等于圆柱的轴长，宽等于半径。使用柱面选项定义的矩形与使用四边形选项定义的不同，前者可以在区域中任意的定位，而后者的长和宽必须与坐标轴一致。

你可以在适当的空框键入确切的坐标，也可以使用鼠标选择网格或解场显示的位置。区域被定义之后，每个在指定区域内部/外部有一个质心的单元都被标记为细化。

5.6.2　区域适应的例子

图 5.6.1 中的网格是为求解围绕一个襟翼翼型的流动而建立的。网格在接近翼型表面的地方非常细以便粘性影响的区域可能被解决。我们知道流动分离发生在襟翼的吸气表面，然而，网格在远离翼型时增长的非常快，以至于不能很好的预知流动分离。要解决这个问题，需对襟翼周围的圆形区域（用鼠标指针选取）内的网格进行适应。适应后的网格如图 5.7.2 所示。注意，当区域适应被执行时，关于适应的最小单元体积被限制（在第 5.10 节中有介绍）以阻止表面附近非常小的单元被进一步细分。

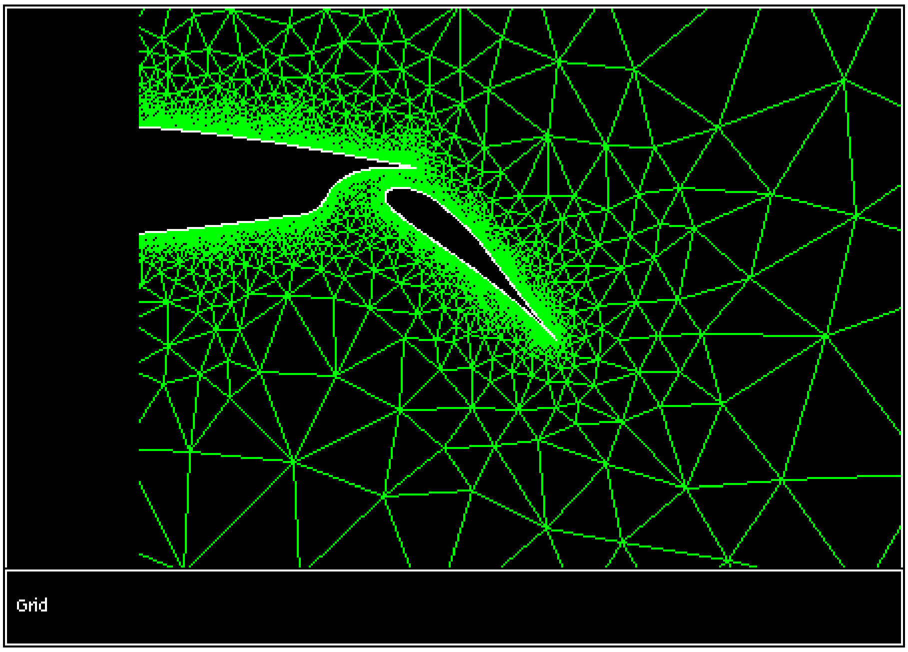


图 5.6.1：适应前的襟翼的翼型网格

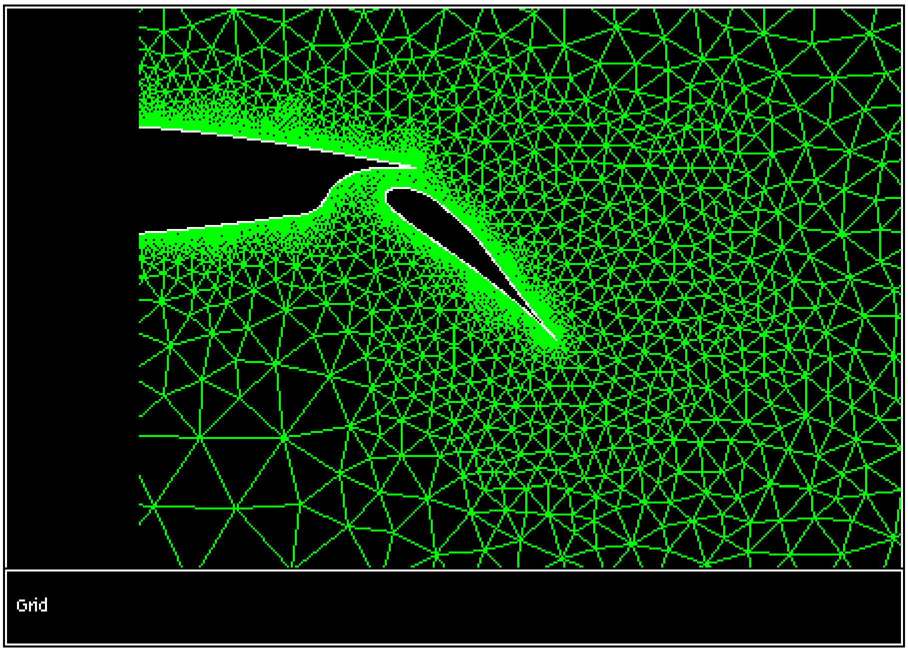


图 5.6.2：区域适应后的襟翼的翼型网格

5.6.3　执行区域适应的步骤

在 Region Adaption 面板中执行区域适应。

A → Region

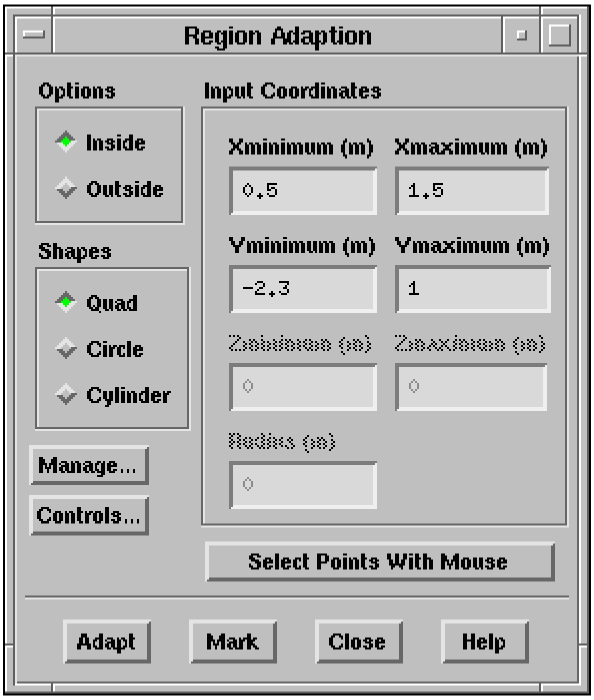


图 5.6.3：Region　Adaption 面板

执行区域适应的一般步骤如下：

1．在 Region Adaption 面板中，选择 Inside 或 Outside 选项。

若选择 **Inside**，则在规定的区域内有质心的单元将被标记或细分。

若选择 **Outside**，则在规定的区域外有质心的单元将被标记或细分。

2．规定区域的形状。在 2D 中，可以选择一个 Quadrilateral，Circle，或 Cylinder。在 3D 中，可以选择一个 Hexahedron，Sphere，或 Cylinder。

3．通过把值键入面板或使用鼠标来定义区域。在面板中如下输入：

要定义一个六面体或四边形，则需要输入定义盒子对角线的两个点的坐标：对一个六面体，（Xminimum, Yminimum, Zminimum）和（Xmaximum, Ymaximum），或对一个四边形，（Xminimum, Yminimum）和（Xmaximum, Ymaximum）。

要定义一个球或圆，则需要球的球心坐标（Xcenter, Ycenter, Zcenter）或圆心坐标（Xcenter, Ycenter），及半径。

要定义一个圆柱，则在 3D 中，需要输入定义圆柱轴的坐标的最小值和最大值一（X-Axis Min, Y-Axis Min, Z-Axis Min）和（X-Axis Max, Y-Axis Max, Z-Axis Max），或在 2D 中，输入（X-Axis Min, Y-Axis Min）和（X-Axis Max, Y-Axis Max），不论 2D 还是 3D，都要输入圆柱的 Radius。（在 2D 中，圆柱的轴长和半径分别是矩形的长和宽。）

要使用鼠标定义区域，则点击 Select Points With Mouse 按钮。使用鼠标探针（缺省时是鼠标左按钮），你可以从一个网格和解场的展示中选择输入坐标。在你选择了点之后，值将被自动地下载到面板的适当的框中。在标记或适应之前，你可以对你想修改的值进行编

辑。

要定义一个六面体或四边形，你可以按任意顺序选择对角线的两个点。

要定义一个球或圆，首先选择质心的位置，然后选择一个位于这个球或圆上的点（即，这点表示远离轴的半径）。

要定义一个圆柱，首先选择定义圆柱轴的两个点，然后选择一个点，它是远离轴的半径。

4.（可选择的）如果你想设置任何适应选项（在第 5.10 节有介绍），点击 **Controls…** 按钮打开 **Grid Adaption Controls** 面板。

5．点击 **Mark**，这样通过把单元放进一个适应记录而为细化给单元做标记，或着点击 **Adapt** 立即执行细化。

5.7　体积适应

如第 5.1 节所述，一个网格中的单元体积渐进的变化对于解的精确性及收敛性是非常有益的。如果这种网格的建立或适应过程导致一个网格内没有这种特性，那么可以在单元体积或单元和它相邻的单元之间的体积变化的基础上，通过使用选项为细化的体积适应来改进网格。

5.7.1　方法

基于体积的大小标记或细分网格被经常用来切除大单元或全局地细化单元。其过程是标记体积大于规定的临界值的任何单元为细化。

基于单元体积内的变化的标记或细分被用来改进网格的光滑性。其过程是标记体积大于规定的临界值的任何单元为细化。体积变化是通过轮循过面和比较单元邻域与面的比率而计算的。例如，在图 5.7.1 中，比率$V1/V2$和比率$V2/V1$与临界值做比较。如果$V2/V1$大于临界值，那么C被标记为细化。

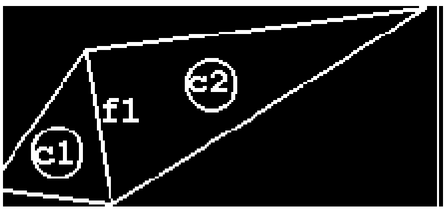


图 5.7.1：作为邻近一个面的单元体积的比率，体积变化被计算

5.7.2 体积适应的例子

图 5.7.1 是为计算一个湍流喷射而建立的网格。在 TGrid 中使用局部细化建立了一个网格，在射流区域中网格是比较细的，而在其它区域则比较粗糙。在射流边上的单元体积有一个急剧的变化。

要改进网格，使用体积适应来细化网格，其中体积适应有一个关于单元体积的变化最不应超过 50% 的标准。对适应的单元体积变化的最小值也有限制。体积适应之后的网格，经过光滑和交换，所得的网格如图 5.7.3 所示。从图中可以看到，射流内的细化区域与周围的网格之间的界面不再那么明显。

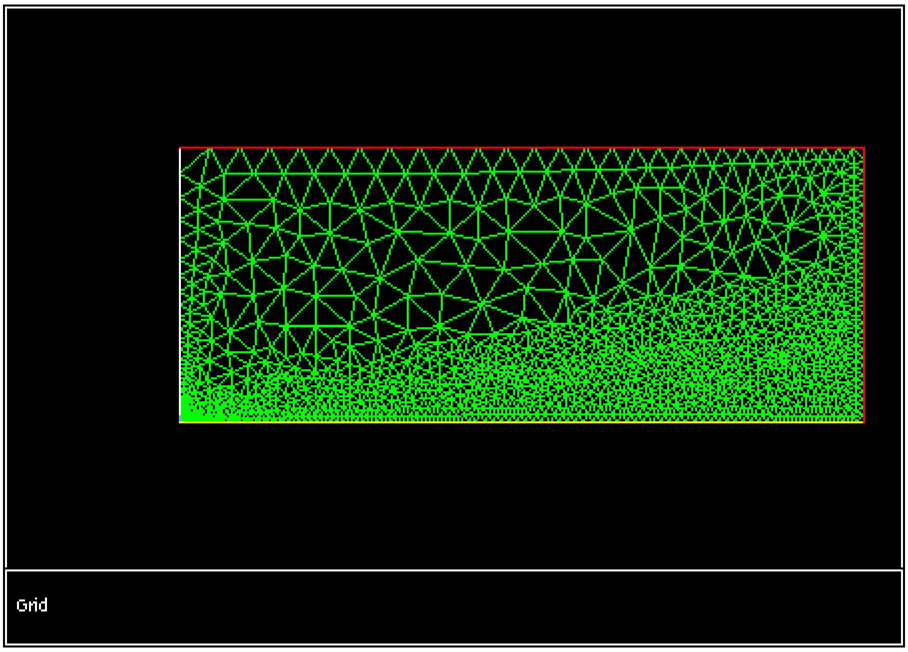


图 5.7.2：适应前的射流网格

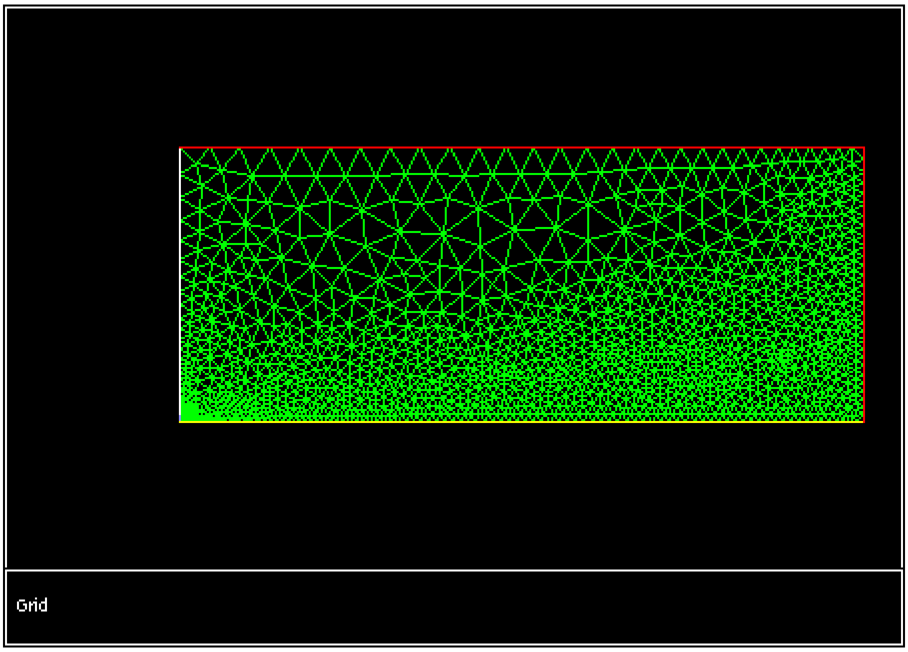


图 5.7.3：基于单元体积的变化的体积适应之后的射流网格

5.7.3 执行体积适应的步骤

在 Volume Adaption 面板（如图 5.7.4 所示）中执行体积适应。

A → Volume

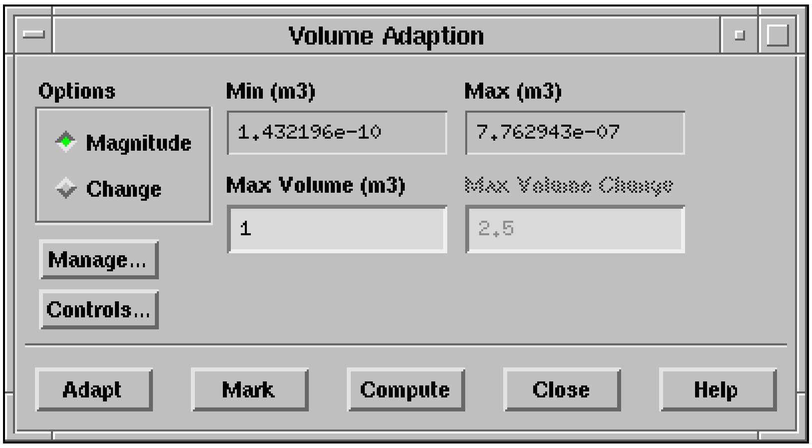


图 5.7.4：Volume　Adaption 面板

执行体积适应的一般步骤如下：

1．在 **Volume Adaption** 面板中，通过选择 **Magnitude** 或 **Change** 选项而规定在体积大小的基础上还是在体积变化的基础上进行适应。

2．点击 Compute 更新 Min 和 Max 框中的值。它们将显示单元体积或单元体积变化的范围（即最小值与最大值），只有哪种的范围当然要取决于第一步的选择。

3．设置 Max Volume 或 Max Volume Change 的值。如果基于体积 Magnitude 进行适应，那么体积大于 Max Volume 的单元将被标记或细化。如果基于体积变化进行适应，那么体积变化大于 Max Volume Change 的单元将被标记或细分。

4．（可选择的）如果你想设置任何适应选项（在第 5.10 节有介绍），点击 **Controls…** 按钮打开 **Grid Adaption Controls** 面板。

5．点击 **Mark**，这样通过把单元放进一个适应记录而为细化给单元做标记，或着点击 **Adapt** 立即执行细化。

5.8　y+和 y*适应

FLUENT 为湍流近壁模型（标准的壁面函数，非平衡的壁面函数，和增强的壁面处理）提供了三种选项。

当建立网格时，由于测量近壁分解的需求往往比较困难，y+和 y*适应能够使你在解的过程中沿壁面适当地细化或粗化网格。如果你正在使用增强的壁面处理，你将对 y+进行适应；如果你正在使用壁面函数，你既可以对 y+也可以对 y*进行适应。

5.8.1　方法

方法是计算关于在规定的粘性壁面区域上的边界单元的 y+或 y*，定义 y+或 y*所允许的最小值和最大值，及标记或适应合适的单元。y+或 y*的值低于所允许的最小临界值的单元将被标记为粗化，y+或 y*的值高于所允许的最大临界值的单元将被标记为细化（除非粗化或细化没有被激活）。

5.8.2　y+适应的例子

图 5.8.1 给出了一个管道流动的网格，其中顶部边界是壁面，底部边界是一个对称平面。在一个初始解之后，确定在壁面边界上的单元的 y+值太大，那么 y+适应被用来细分单元。适应后的网格如图 5.8.2 所示。从图中可以看出，在细化过程中单元沿壁面边界的高已被降低了。然而，细化后的单元尺寸分布没有原来网格的整齐，它是 y+适应所产生不利效果。

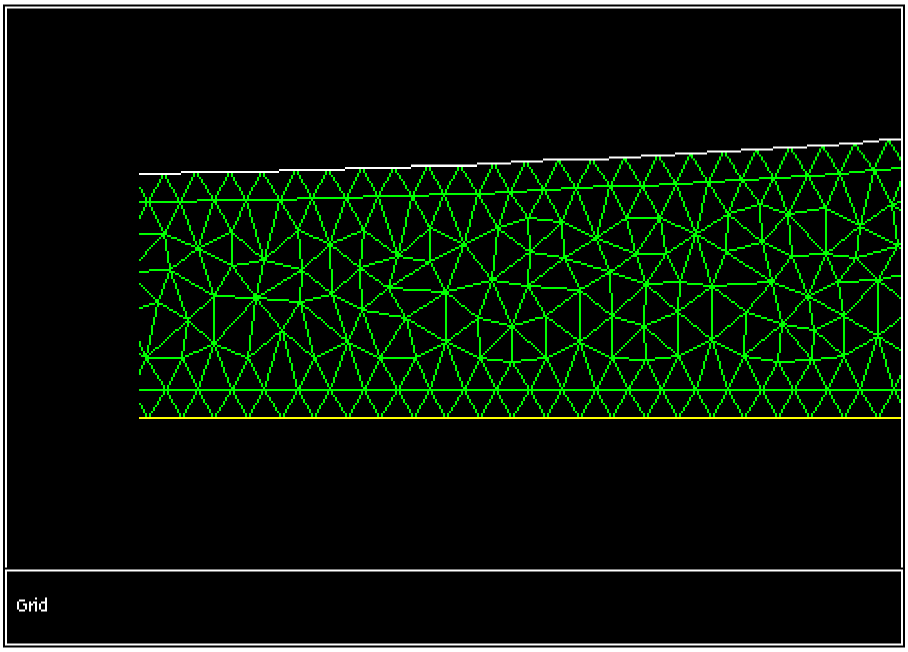


图 5.8.1：适应前的管道流网格

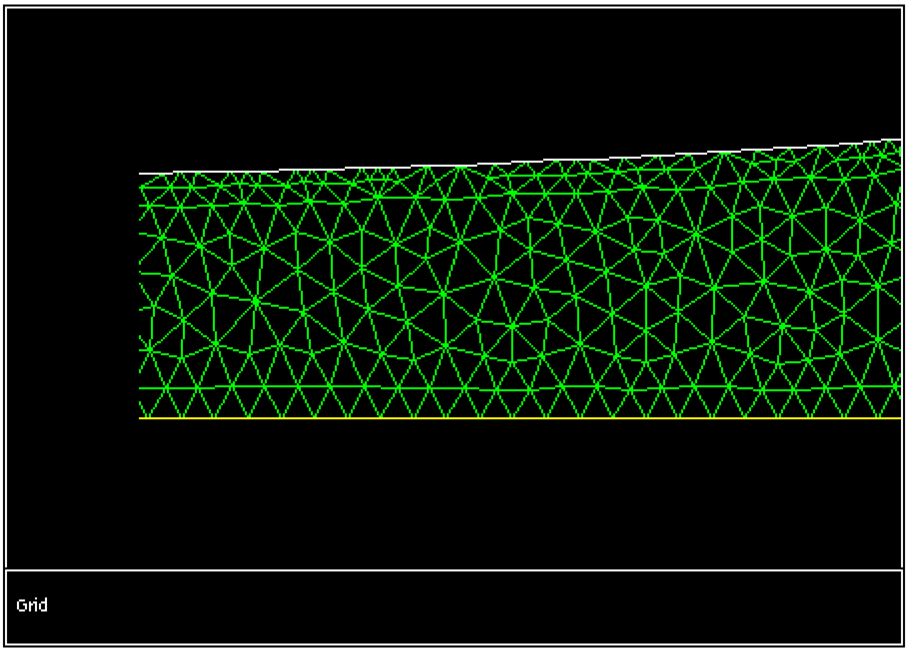
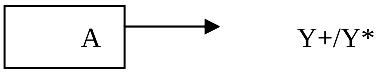


图 5.8.2：y+适应后的管道流网格

5.8.3 执行 y+或 y*适应的步骤

在 **Y+/Y*Adaption** 面板中执行 y+或 y*适应（如图 5.8.3 所示）。



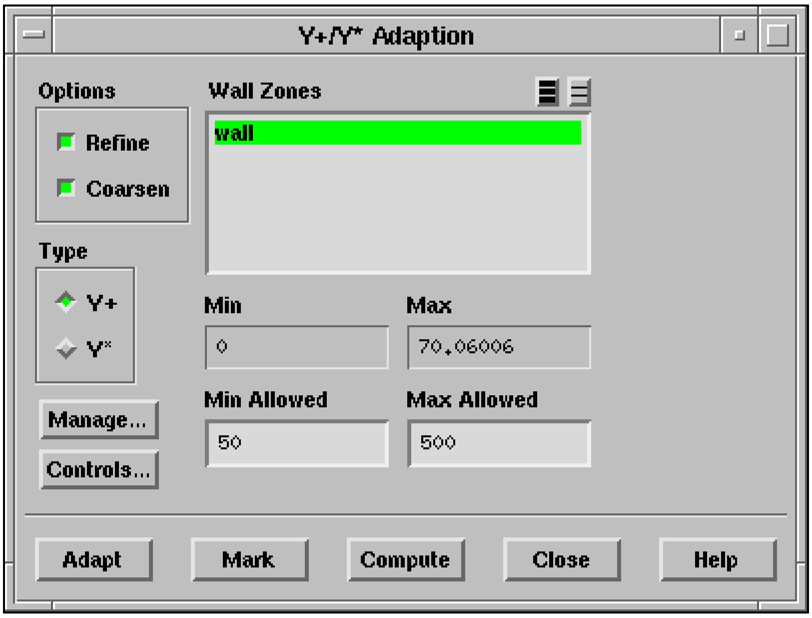


图 5.8.3：Y+/Y* Adaption 面板

执行 y+或 y*适应的一般步骤如下：

1．在 **Y+/Y* Adaption** 面板中，选择适应的类型 **Y+**或 **Y***。（若使用增强的壁面处理，则选择 **Y+**；若使用壁面函数，则两种类型皆可选择。）

2．在 **Wall Zones** 列中选择壁面区域，其中壁面区域上的边界单元是你相对其进行标记或适应的。点击 **Compute** 更新 **Min** 和 **Max** 框中的值。注意，所显示的值是关于所有壁面区域的最小值和最大值，而不只是那些所选壁面区域的。

3．设置 **Min Allowed** 或 **Max Allowed** 的值。y+或 y*的值小于 Min Allowed 的值的单元将被粗化或标记为粗化；y+或 y*的值大于 Max Allowed 的值的单元将被细化或标记为细化。

注意，如果你正在使用悬挂节点适应（缺省时），那么你将不能建立一个比原来的网格更粗的网格。这时，你必须使用正则适应。还需要注意的是，正则粗化仅适用于 2D 或轴对称几何。相关内容在第 5.10 节有介绍。

4．（可选择的）如果你想设置任何适应选项（在第 5.10 节有介绍），点击 **Controls…** 按钮打开 **Grid Adaption Controls** 面板。

5．点击 **Mark**，这样通过把单元放进一个适应记录而标记单元为适应（细化/粗化），或着点击 **Adapt** 立即执行适应。

5.9　管理适应记录

你可以操作、删除和展示通过标记单元为适应而建立的适应记录。由于这些记录将被用来调整网格，因此操作它们的功能为你提供了对适应过程的额外控制。

在 Manage Adaption Registers 面板中执行适应记录的管理（如图 5.9.1）。

A → Manage



图 5.9.1：Manage Adaption Registers 面板

（你也可以在任何适应面板中通过点击 **Manage…**按钮打开这个面板。）

一般来说，想把适应局限在一个特定的区域（适应标记）和/或建立一个更准确的误差指示器是激发建立混合适应函数的因素。FLUENT 提供了几个帮助建立混合适应函数的工具。第一，你可以使用几何的和/或解的矢量信息来建立初始的适应记录。在建立了适应记录之后，你可以操作这些记录和与它们相关的细化和粗化标记。改变记录的类型和/或把记录结合起来以建立所需要的混合函数，从而达到操作记录的目的。标记是通过使用

Exchange，Invert，Limit，和 Fill 来操作的。最后，你可以删除，展示，和最重要的一点即适合混合适应函数。

5.9.1　操作适应记录

用于修改和操作适应记录的基本的工具有三个：改变类型，合并和删除。

改变记录的类型

目前，有两种类型的记录：适应记录和掩饰记录。目前的适应函数通过 GUI 被访问，它和文本界面产生适应记录ー即，指定被标记为细化或粗化的单元的记录。这些记录可以被转换成掩饰记录，不过，要通过改变它们的类型实现。不像适应记录，掩饰记录仅保持两种状态：ACTIVE 和 INACTIVE。如果适应记录被转换成一个掩饰的，那么被标记为细化的单元使 ACTIVE，其它的单元都是 INACTIVE；即不考虑被标记为粗化的单元。一般地，被转换成掩饰的适应记录是由专门把单元标记为细化的适应函数所产生的那些记录，例如区域或各向同性适应函数。适应和掩饰记录的结合的方式也是它们的主要的不同之处。

把一个或多个记录从适应改变成掩饰类型（反之亦然）的步骤如下：

1．在 **Registers** 列中选择记录。

2．点击 Register Actions 下面的 Chang Type。

新类型的记录作为 **Type** 将出现在 **Register Info** 下面的框里。你可以单独地选择每个记录来查看它当前的类型。

合并记录

在每个适应记录被建立和适当地修改之后，这些记录被合并起来以建立混合适应函数。用下面的方式可以合并任意数目的记录：

把所有的适应记录合并成一个新的适应记录。

把所有的掩饰记录合并成一个新的掩饰记录。

把新适应记录和新掩饰记录合并起来。

用下面的方式可以合并任意数目的适应记录：

如果单元在任意的记录中被标记为细化，那么在新记录中标记单元为细化。

如果单元在任意的记录中被标记为粗化，那么在新记录中标记单元为粗化。

合并掩饰记录的方式类似于细化标记：如果任何单元被标记为 ACTIVE，那么在新的记录中的单元被标记为 ACTIVE。

最后，在一个适应和掩饰记录的合并中，仅有在掩饰记录中被标记的单元可以在合并的记录中有一个适应标记。

例如，基于压力梯度建立一个适应函数可能在整个解域中产生标记为细化和粗化的单元。如果那时这个记录与一个掩饰记录合并，其中这个掩饰记录是从一个球内部的被标记的单元建立的，那么仅有在球内部的单元将在新记录中被标记为细化或粗化。

掩饰的影响依赖于它们被应用的顺序。

合并两个或多个记录的步骤是：

1．在 **Registers** 列中选择记录。

2．点击 Register Actions 下面的 Combine 按钮。

所选的记录将保持完好无缺，来自合并的记录将被添加到 **Registers** 列中。在某些例子中，三个新的记录可能被建立：适应记录的一个合并，掩饰记录的一个合并，和两个合并的记录的一个合并。

删除记录

适应记录被删除的数目是任意的。删除记录的主要原因是放弃不需要的适应记录，以减少可能的混乱，同时，删除这些无价值的记录可以降低生成不需要结果的潜能。另外，只有 32 个适应记录可以同时存在。所以，你可能需要删除不需要的记录以为新的腾出空间。

永久的删除一个或多个记录的步骤是：

1．在 **Registers** 列中选择记录。

2．点击 Register Actions 下面的 Delete 按钮。

5.9.2 修改适应标记

适应标记是用来标记一个单元应当被细分、粗化还是中立（既不细分也不粗化）的标识符。目前，修改适应标记的基本工具有四个：**Exchange**，**Invert**，**Limit**，和 **Fill** 操作。

Exchange 操作把所有标记为细化的单元改变成标记为粗化的单元，把所有标记为粗化的单元改变成标记为细化的单元。一般地，这个操作适应于只有细化标记的适应记录。例如，交换操作可以被用来粗化一个矩形区域。首先，你建立一个把一矩形区域的单元标记为细化的适应记录。然后，使用 **Exchange** 操作修改单元标记，从而建立一个矩形区域，且其中的单元被标记为粗化。

Invert 操作仅可以用于掩饰记录。这个操作使得所有 ACTIVE 标记的单元转化到 INACTIVE，所有 INACTIVE 标记的单元转化到 ACTIVE。例如，如果你生成了一个定义一圆形区域的掩饰，你可以使用 **Invert** 操作快速地修改这个掩饰以定义圆外部地区域。

Limit 操作对所选择的适应记录应用现在的适应体积限制。一般地，你使用这个操作确定现在的限制对适应过程的影响。你可以通过仅对细化大单元设置限制来使用体积限制以建立一个更均匀的网格。在所有的单元达到一相同尺寸之后，为获得要想的分解，你可以继续细化过程。

最后，**Fill** 操作把在适应记录中没有被标记为细化的单元标记为粗化的。如果你正则把多个记录合并成一个新的记录，那么也许你想使用 **Fill** 操作。当你合并记录的时候，一个单元当且仅当在所有的记录中被标记为粗化的时候才将被标记为粗化。如果你建立一个适应记录，且它有一个仅把单元标记为细化的操作，但你不想禁止粗化，那么你应当在把这个记录与任何其它的记录合并之前使用 **Fill** 操作。

修改适应标记的步骤是：

1．在 **Registers** 列中选择记录。

2．点击 Mark Actions 下面的 Exchange，Invert，Limit 或 Fill 按钮。

5.9.3　展示记录

查看单元的标记在建立混合适应的过程中往往是很有用的。你可以策划一个在单元的质心的标记和/或一个单元的接线框以查看单元的状态。缺省时，标记为细化的单元被涂以红色，而粗化的则被涂以蓝绿色。另外，在一个掩饰记录中，标记为 ACTIVE 的单元也被涂以红色。这些是标记为适应的单元，但添加到网格中或从网格中减去的单元，其最后的个数将取决于适应的限制和网格的特性。

展示一个记录的步骤是：

1．在 **Registers** 列中选择记录。

2．若需要，点击 **Options…**，设置任何展示选项（下面有介绍）。

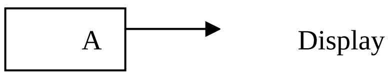
3．点击 **Display** 按钮。

适应的展示选项

适应记录展示的各个方面都可以被修改，例如接线框的可视性，留下阴影、标记的可视性，颜色，大小，符号，以及所展示的表面或面网格。

适应记录展示的能力允许你查看被标记为适应的单元。你可以使用标记和/或接线框展示在一个适应或掩饰记录里的被做标记的单元。标记是被放在单元质心的一个符合，包括细分标记和粗化标记。你可以改变这些标记的符合、颜色和大小、一个连接框由三角形或四面体的边组成，它的颜色和各自的标记颜色相同，并且它还可以用此颜色填充。

最后，网格的部分可以用标记符号或接线框画出，这有助于估测标记的单元的位置。



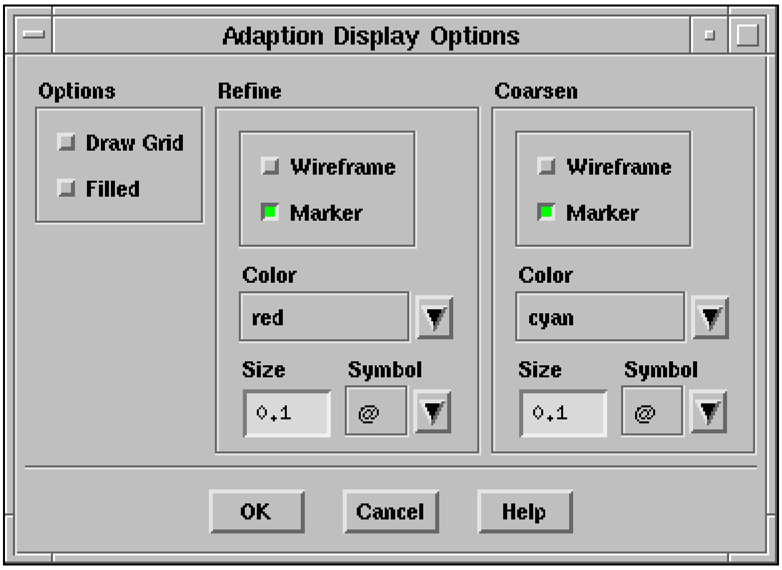


图 5.9.2：Adaption Display Options 面板

（你也可以通过点击 Manage Adaption Registers 面板中的 Options…按钮打开这个面板。）

要激活或不激活关于被标记为细化/粗化的单元的接线框的展示，请打开或关掉 **Refine** 和/或 **Coarsen** 下面的 **Wireframe** 选项。若要填充接线框，则打开 **Filled** 选项即可。

要激活或不激活关于被标记为细化/粗化的单元的标记的展示，请打开或关掉 **Refine** 和/或 **Coarsen** 下面的 **Marker** 选项。如果你使用标记，那么你可以在 **Size** 的框里规定它们的大小，在 **Symbol** 下面的列中规定它们的符合。

要改变细化或粗化的标记/接线框的颜色，则在 **Refine** 或 **Coarsen** 下面的 **Color** 下面的列中选择一个新颜色。缺省时，细化标记/接线框是红色，粗化标记/接线框是蓝绿色。

要展示在记录展示中的网格的一部分，则激活 **Draw Grid** 选项。**Grid Display** 面板将自动地出现，你可以在这里设置网格展示的参数。当你在 **Manage Adaption Registers** 面板中点击 **Display**，在 **Grid Display** 面板中定义的网格展示将被包含在记录展示中。

5.9.4 适应记录

调整网格的主要目的是有效地提高解的精确度。这些记录工具为你提供了建立混合的适应函数能力。最后，用户化的适应函数被用来管理细化和粗化的网格。

执行适应的步骤如下：

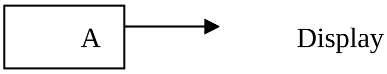
1．选择 **Registers** 列中的记录。

2．点击 **Adapt** 按钮。

5.10　适应性控制

FLUENT 允许改变适应的类型—即从悬挂节点到正则，反之亦然，或者对单元区域、可以被调整的单元的尺寸、以及从适应过程中产生的单元数目给予约束。你也可以修改梯度函数中的体积权重的强度，限制细化和/或粗化的适应过程，及在正则粗化过程中，控制哪些节点适合被从网格中删除。

控制适应的这些方面的所有参数是在面板 **Grid Adaption Controls**（如图 5.10.1 所示）中设置的。



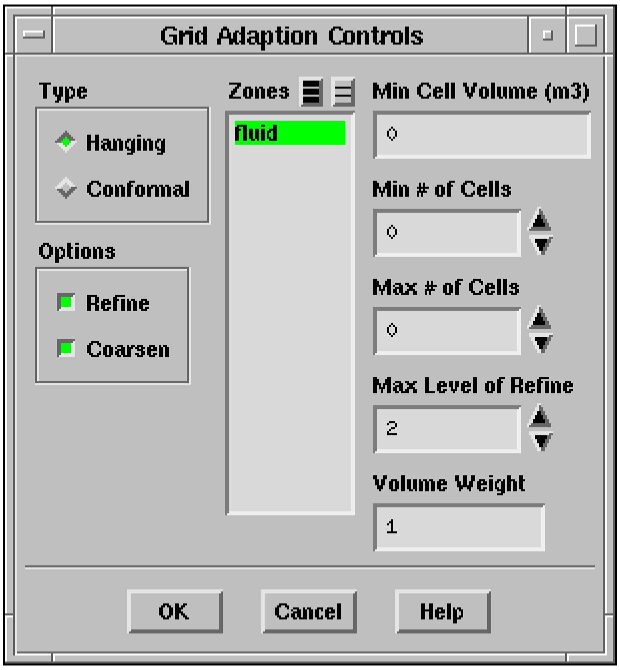


图 5.10.1：Grid Adaption Controls 面板

（也可以在任何一个适应面板中通过点击 **Controls...**而打开这个面板。）

控制适应的类型

你可以选择使用悬挂节点适应还是正则适应。你也可以限制适应的过程，其中包括经过细化增加网格的分解，经过粗化消减网格密度。

要使用悬挂节点适应（缺省），则选择 Type 框中的 Hanging，要使用正则适应，则选

择 Conformal。关于悬挂节点和正则的适应程序在第 5.2 节中有具体介绍。

要激活/不激活细化，则打开/关闭 **Refine** 选项。

要激活/不激活粗化，则打开/关闭 **Coarsen** 选项。

借助域限制适应

你可以把适应过程限制到规定的单元域。一般地，构成流体和固体区域的单元有非常不同 分解要求和误差指示器。通过把适应限制到一个特定的单元域，你可以使用不同的适应函数以建立最佳网格。

要把适应限制到一个特殊的单元域，则在 **Zones** 列中选择单元域（你想在这些单元域中执行适应）。缺省时，适应将在所有单元域中被执行。

借助单元体积或体积重量限制适应

最小单元体积的界限把细化过程限制在体积大于界限的单元中。你可以使用它在较大的单元上开始细化过程，逐渐地降低界限以建立一个均匀单元尺寸的分布在 **Min Cell Volume** 的框里设置这个界限。注意，对于一个 2D 轴对称问题，在这个框中输入的则是最小单元面积。

另外，梯度体积重量可以被修改。一个零值排除了体积权重的情况，1 使用整个体积，介于 0 到 1 的值则依比例决定体积权重。在 **Volume Weight** 的框中设置这个值。

限制单元的总数

你可以在 **Max # of Cells** 的框中设置网格中所允许的最大单元总数。在 **Min # of Cells** 的框中设置网格中的最小单元个数。缺省时的零值对单元数目没有限制。

在悬挂节点适应过程中控制细化层面

在非正则适应过程中，细化被用来切开单元。通过设置 **Max Level of Refine**，你可以控制此细化层面的个数。对于大多数问题，缺省值 2 是一个好的出发点。如果这个值不够，那么你可以增加这个值，但是请记住，即使在一个 2D 问题中，在适应的区域，缺省值 2 也能够以 16 的因数增加单元的个数。一个 0 值使得这个参数极大：在这种情况下，你应当对 **Min Cell Volume** 使用一个适当的限制。请看第 5.2.2 节关于悬挂节点适应的更详细的相关内容。

在正则粗化过程中控制节点的切除

通过修改节点切除的记号，你可以控制粗化过程中的节点切除。节点切除记号仅适用于正则适应。对于悬挂节点适应，仅细化节点能够在粗化过程中被切除，及它们一直被删除。

为保证区域边界的原来形状被保持，通常仅由细化引进的节点被切除。例如，考虑一

个矩形区域的网格。如果在矩形边界上的一个节点被切除，区域的形状不改变，但是如果矩形的一个角上的节点被删除，形状则从一个四边形变成一个五边形。

通过改变节点切除记号，你可以修改缺省行为。大部分普通的修改允许原始的内部节点的切除。除去内部节点不破坏边界形状。事实上，如果原始网格在一个区域中有实质的分解，而在物理特性上没有或有很小的变化，那么这是非常有益的。

缺省时，仅允许细化节点的切除，正如下面所指示的：在 Grid Adaption Controls 面板中，激活 Boundary-Refined，Internal-Refined 和 Periodic-Refined，不激活 Boundary-Original Internal-Original 和 Periodic-Original。如果你想使这些类型的节点的切除处于不活动状态，那么你可以关掉相关的控制按钮。类似地，如果你想激活它们的切除，打开相关的控制按钮。

5.11 用光滑和交换的方式改善网格

通常情况下，光滑和面交换会提高最后数值网格的质量，所以说它们是补助网格适应的工具。光滑重新配置节点和面的交换修改单元的连通性，从而在质量上取得改善。

！面交换仅对三角形和四面体单元的网格适应。

使用 **Smooth/Swap Grid** 面板（如图 5.11.1 所示）来操作光滑和交换。

A → Smooth/Swap…

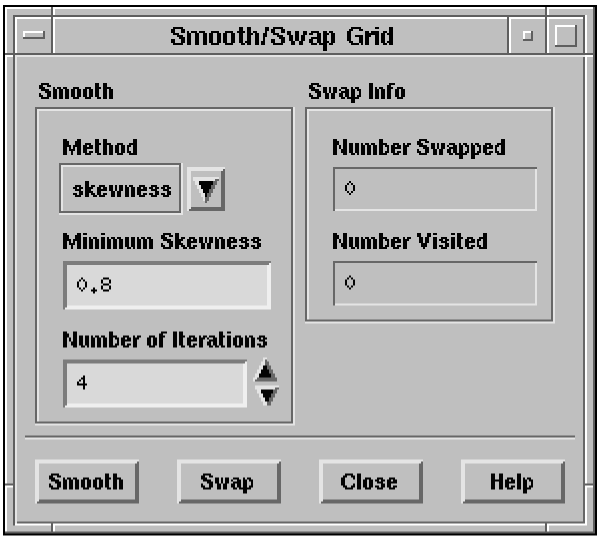


图 5.11.1：Smooth/Swap Grid 面板

5.11.1 光滑

在 FLUENT 中关于光滑的方法有两种：Laplace 光滑和基于偏斜的光滑。第一种方法可以用于各种类型的网格，而第二种方法仅适用于三角形/四面体网格。

！建议对三角形和四面体网格一直使用基于偏斜的光滑方法；而把 Laplacian 方法仅使用于四边形和六面体网格。

Laplacian 光滑

当你使用 Laplace 方法时，一个 Laplacian 光滑操作被应用于非结构网格以重新配置节点。新节点的位置是它周围节点位置的平均。松弛因子（介于 0.0 到 1.0 的一个数）乘上计算所得的节点位置的增量。松弛因子取 0 值时节点没有移动，取单位 1 时则节点的移动等于节点位置的整个增量。对于四边形单元的一个典型的结构，图 5.11.2 给出了新节点的位置。

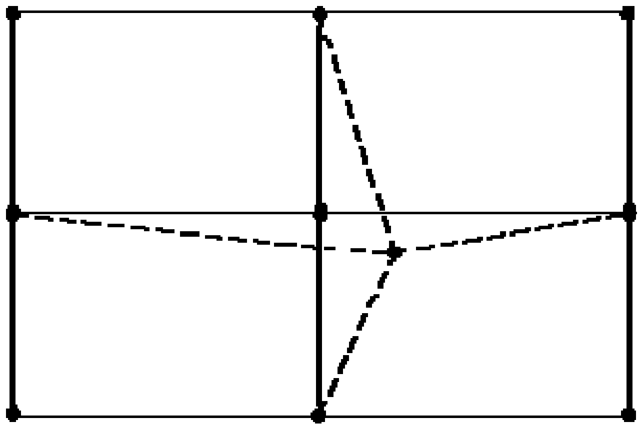


图 5.11.2：关于节点位置的光滑操作的结果图（虚线表示原来的网格，实线表示最后的网格）

重配置策略倾向改善网格的歪斜，但通常对节点的聚类放松约束。在极端的情况下，目前的操作可能建立越过边界的网格线，从而建立了负单元体积。特别是你使用一个大的松弛因子执行了多个光滑操作，那么这最有可能发生在接近急剧拐角或粗糙解决的凸起的拐角的地方。图 5.11.3 是未松弛的光滑迭代之前的一个初始四面体网格，通过它建立互相超出的网格线（如图 5.11.4 所示）。

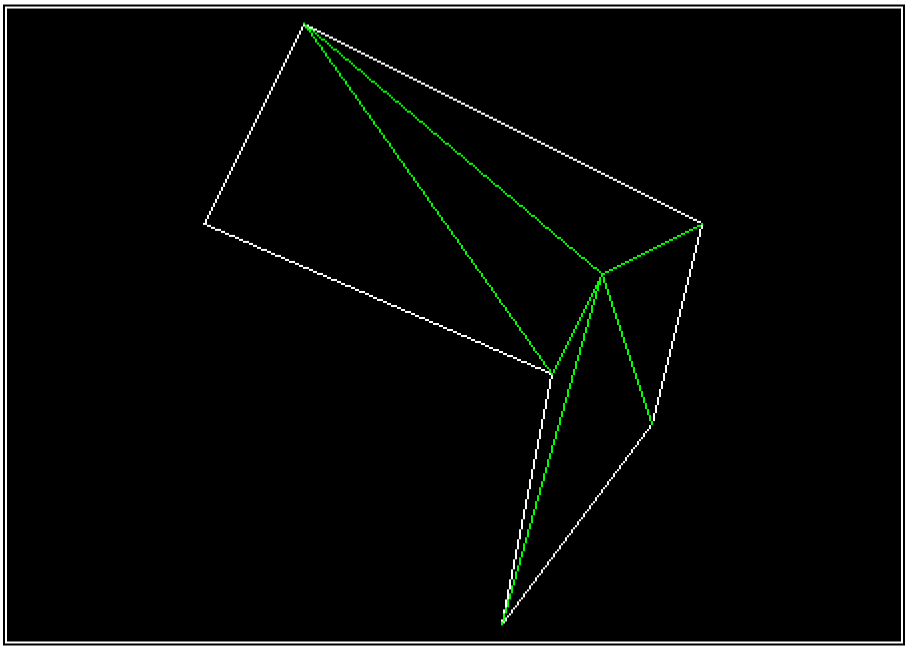


图 5.11.3：光滑操作前的原始网格

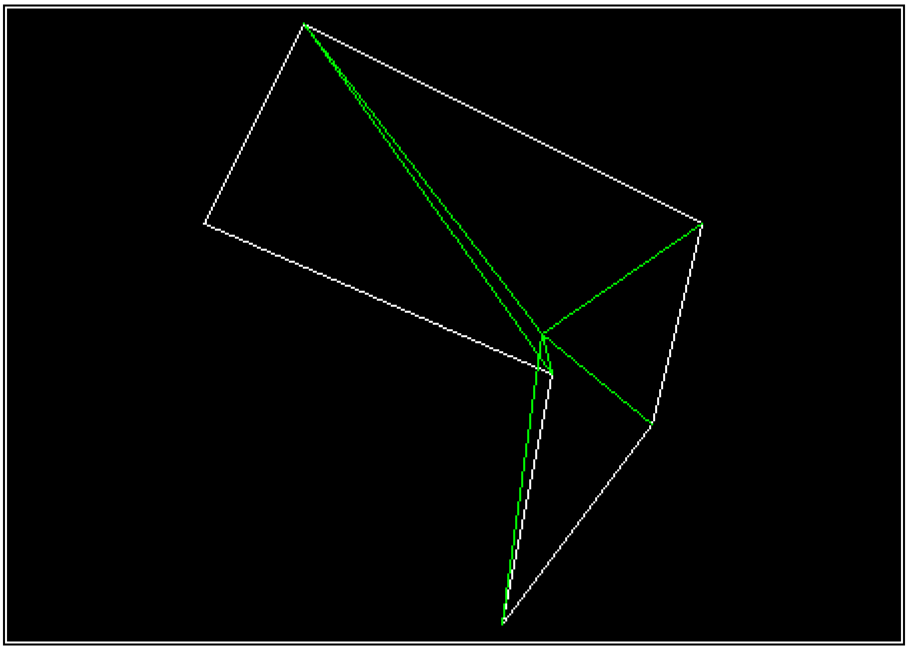


图 5.11.3：网格光滑会导致网格线交叉

缺省的光滑参数被安排来以最小的反面效果改进网格质量，但在光滑网格之前，你应当记得保存一个 case 文件。如果在一个二维问题中，你应用了一个保守的松弛因子，并且

从一个高质量的初始网格出发，那么以后的失败是由光滑极低造成的。然而，网格拓扑的恶化在三维问题中发生的要频繁的多，特别是四面体网格。对于三角形和四面体网格建议用基于歪斜的光滑操作。

光滑操作也可以被重复的应用，但是随着光滑扫过次数的增加，节点有被从边界拉走的趋势，网格倾向于丢失任何聚集的特性。

Laplacian 光滑的步骤

执行 Laplacian 光滑的步骤如下：

1．在 Smooth/Swap Grid 面板中（如图 5.11.1 所示），选择 Smooth 下面的 Method 下面列中的 laplace。

2．在 **Relaxation Factor** 框中，设置因子。因子越低，在节点运动中降的越多。

3．在 **Number of Iterations** 框中，规定光滑相继扫过的网格个数。缺省值为 4。

4．点击 **Smooth** 按钮。

基于歪斜的光滑

当你使用基于歪斜的光滑时，FLUENT 对网格应用一个光滑操作，重新配置内部节点以降低网格的最大歪斜。FLUENT 将努力移动内部节点以改善这些大于规定的“最小歪斜”的单元的歪斜。这个过程非常耗费时间，所以你应当只在高歪斜的单元上执行光滑。通过几次执行光滑节点，可以获得改善的结果。如果移动一个节点导致最大歪斜增大，那么内部控制将阻止它被移动，但是当一个高歪斜的节点被移动时，某些单元的歪斜将被增加是普遍现象。所以，你可能看到在最大歪斜减小的同时平均歪斜增大。**!** 你应当仔细考虑这样做是否对改善网格有意义。仅对带有很高歪斜（例如，0.8 或 0.9）的单元执行光滑也许会缓解对平均歪斜所产生的反面效果。

! 基于歪斜的光滑仅适用于三角形和四面体网格。

基于歪斜的光滑的步骤

执行基于歪斜的光滑的步骤如下：

1．在 Smooth/Swap Grid 面板中（如图 5.11.1 所示），选择 Smooth 下面的 Method 下面列中的 skewness。

2．在 **Minimum Skewness** 框中，设置最小的单元歪斜值。FLUENT 将努力移动内部节点以改进歪斜值大于这个值的单元的歪斜情况。缺省时，对于 2D 情况，**Minimum Skewness** 被设置为 0.4，对于 3D 则为 0.8。

3．在 **Number of Iterations** 框中，规定光滑相继扫过的网格个数。缺省值为 4。

4．点击 **Smooth** 按钮。

基于歪斜的光滑应当与面的交换（看第 5.11.3 节）交替。

5.11.2 面的交换

面的交换可以被用来改善一个三角形或四面体网格的质量。

三角形网格

关于三角形网格的方法是使用 Delaunay 圆周测试来判定是否一个面应当被交换，其中两个三角形单元共享这个面。如果一个单元的外接圆不包含第二个单元中的未被共享的节点，那么这对共享一个面的单元满足圆周测试。图 5.11.5 给出了关于满足和不满足圆周测试的两个单元邻居的例子。在圆周测试不满足的情况，对角线或面被交换，如图 5.11.6 所示。

重复使用面交换技术将产生一个强迫的 Delaunay 网格。如果一个网格是 Delaunay 网格，那么它是唯一的一个最大化网格中的最小角的分成三角形。所以，这个分成三角形趋向于等边单元，为给定的节点分布提供了最好的等边网格。

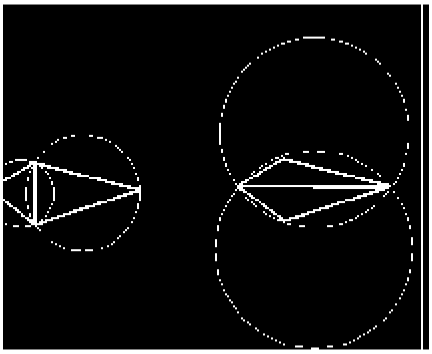


图 5.11.5：满足和不满足圆周测试的单元结构



图 5.11.6：面被交换以满足 Delaunay 圆周测试

四面体网格

对于四面体网格，面的交换包括：寻找三个单元共享一个边的结构和把它们转换成两个单元共享一个面以降低歪斜和单元计数。（看图 5.11.7）

面交换的步骤

执行面交换，点击 **Swap** 按钮直到 **Number Swapped** 的报告为 0。**Number Visited** 表

示关于可能的面交换，被访问和测试的面的总数。

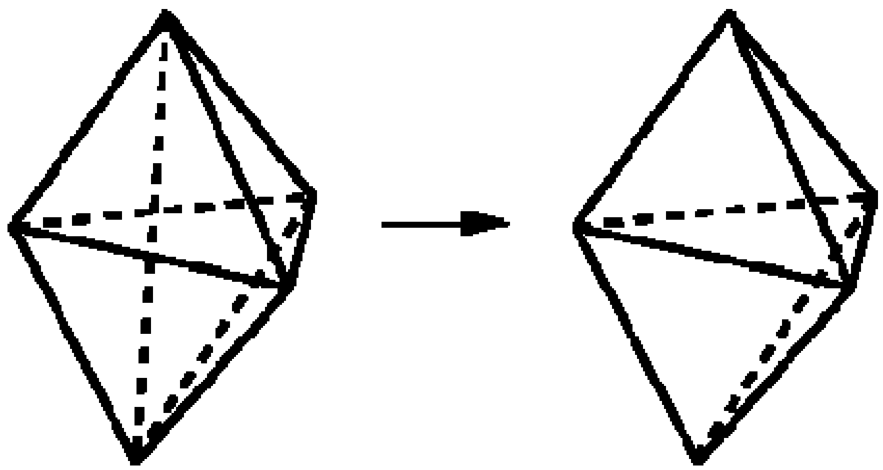


图 5.11.7：3D 面的交换

5.11.3　结合基于歪斜的光滑和面的交换

通常，基于歪斜的光滑应当与面的歪斜交替使用。关于这个程序的指定方法如下：

1．使用一个 Minimun Skewness－对于 3D 是 0.8，对于 2D 是 0.4－执行 4 个光滑迭代。

2．执行交换，直到 Number Swapped 降到 0。

3．对于 3D 网格，把 Minimum Skewness 减小到 0.6，重复光滑/交换的步骤。

第六章 求解技术

本章将讲述 FLUENT 的求解技术，包括数值格式、离散化方法、分离求解器、耦合求解器、多重网格法等内容，以及如何在计算中如何使用这些方法。求解技术是 CFD 软件的核心内容，了解这部分内容，对理解其他部分的内容有很大帮助。

6.1 数值格式回顾

FLUENT 中有两种求解器，即分离求解器（segregated solver）和耦合求解器（coupled solver）。这两种求解器的求解对象是相同的，即它们所求解的控制方程均为描述质量守恒、动量守恒和能量守恒的连续方程、动量方程和能量方程。在考虑湍流和化学反应时，还要加上湍流方程和化学组元方程。它们都用有限体积法作为对计算对象进行离散求解的基础方法，有限体积法的主要工作包括：

- 通过网格划分将空间区域分解成由离散的控制体组成的集合。
- 在控制体上用积分形式构造离散变量的代数方程。
- 将离散方程线化，然后通过求解线化方程获得变量的迭代解。

两种求解器的区别在于它们所使用的线化方法和求解离散方程的方法是不同的。

6.1.1 分离求解方法

分离求解方法即分别求解各个控制方程的方法。由于控制方程是非线性的，因此求解必须经过多次迭代才能获得收敛解。图 6-1 为求解流程图，其过程可概述如下：

（1）流场变量更新。在第一次计算时，变量由初始化过程更新。在随后的计算中，每迭代一次既得到一个更新的解。

（2）用当前压强和质量通量的值求解动量方程，以得到新的速度场。

（3）因为（2）中得到的速度场的数值解无法完全满足连续方程，于是再求解压强修正方程。压强修正方程是由连续方程导出的泊松型方程，求解这个方程可以得到对压强场、速度场和质量通量的修正，进而使连续方程得到满足。

（4）利用前面求出的解，求解湍流方程、能量方程、组元方程和能量方程。

（5）在多相流计算中如果考虑相间干扰，则需要通过求解弥散相轨迹计算得到连续相方程中的源项解。

（6）检验收敛条件是否被满足。如果收敛条件被满足，则停止计算。如果计算没有收敛，则继续迭代过程。

分离算法中采用压强速度耦合算法进行计算，具体格式包括 SIMPLE、SIMPLEC 和 PISO 三种。

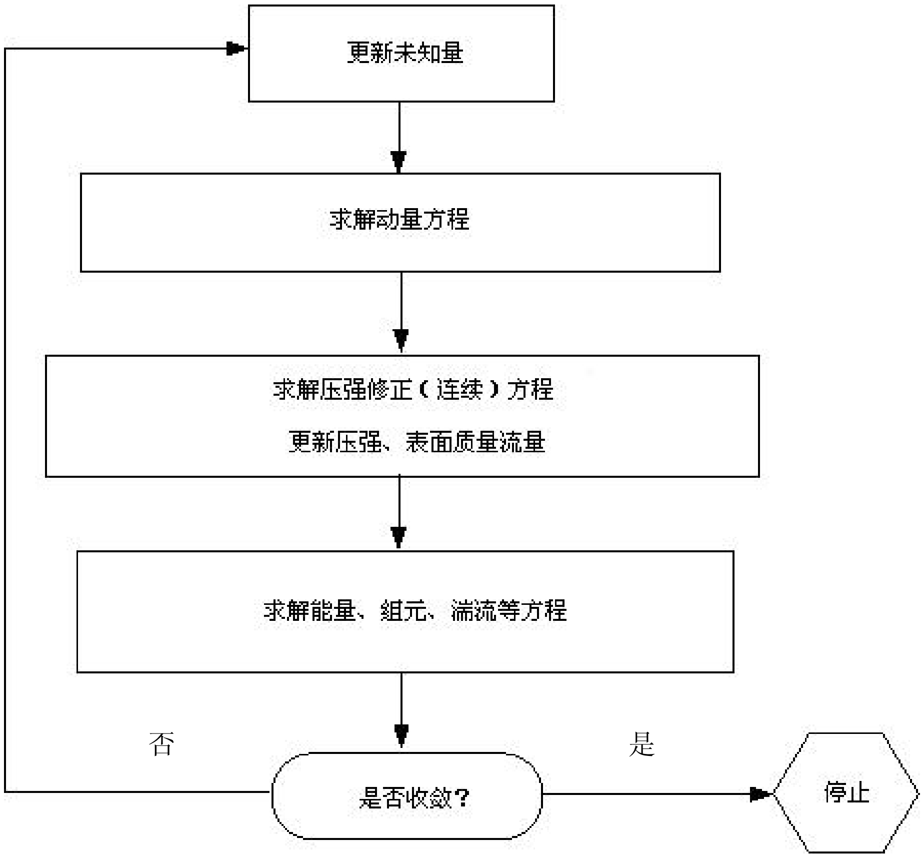


图 6-1 分离算法流程图

SIMPLE 算法的基本策略是用假定的压强场求解动量方程得到边界点上的通量。因为假定的压强场不准确，所以求得的通量必然不能满足连续方程，于是在通量上添加修正项，以使所得通量能够满足连续方程。而通量修正项是压强修正项的函数，因此将修正过的通量代入连续方程，就可以得到一个关于压强修正项的方程。用 AMG 多重网格法求解这个方程可以得到压强修正项的解。在压强修正项的前面乘以亚松弛因子，再与原压强相加就得到一个新的压强场。以这个新的压强场为起点重复上述过程，就形成交替求解压强场、速度场的迭代过程，直到最后得到收敛解，计算既告结束。

SIMPLEC 算法与 SIMPLE 算法的基本思路一致，仅在通量修正方法上有所改进，加快了计算的收敛速度。

SIMPLE 算法和 SIMPLEC 算法在每个迭代步中得到的压强场都不能完全满足动量方程，因此需要反复迭代，直到收敛。PISO 算法针对 SIMPLE 算法中每个迭代步获得的压强场与动量方程偏离过大的问题，在每个迭代步中增加了动量修正和网格畸变修正过程，因此虽然 PISO 在每个迭代步中的计算量大于 SIMPLE 算法和 SIMPLEC 算法，但是由于每个

迭代过程中获得的压强场更准确，所以使得计算收敛得更快，也就是说获得收敛解需要的迭代步数大大减少了。

6.1.2 耦合算法

与分离算法分别求解各个方程相反，耦合算法同时求解连续方程、动量方程和能量方程。在上述流场控制方程被求解后，再求解湍流、辐射等方程，所用方法与分离算法中相同。计算过程也需要经过迭代才能收敛得出最终的解。耦合算法的计算流程可以简单地用图 6-2 表示，大致过程为：

（1）流场变量更新。如果是第一步迭代，则流场变量由初始化过程更新。如果不是第一步计算，则由求解过程得到更新的解。

（2）同时求解连续方程、动量方程、能量方程和组元方程。

（3）在考虑湍流和辐射等模型的时候，求解湍流和辐射方程。

（4）在多相流计算中，如果考虑相间干扰，则通过弥散相轨迹计算求解连续相方程中的源项。

（5）检验计算是否收敛——如果计算已经收敛则停止计算，如果没有收敛则开始新一步的计算。

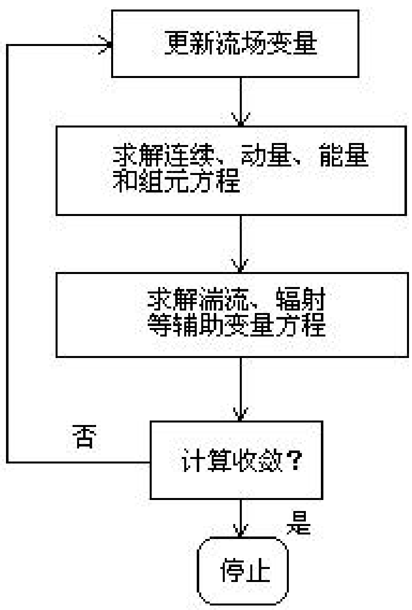


图 6-2 耦合算法流程图

6.1.3 方程的线化：隐式格式与显式格式的对比

流体的控制方程是非线性方程，在数值求解过程中需要将非线性方程在网格单元中化为线性方程，然后再进行求解。所谓隐式格式和显式格式是对方程进行线化和求解的两种不同方式。隐式格式将未知的流场变量（密度、速度、能量等）同已知量之间的关系用方程组的形式加以表达，然后通过求解方程组获得未知变量的值。显式格式则是将未知的流场变量写作已知量的显式函数形式，因此每个变量可以用一个方程单独进行求解。

分离算法中，每个流场变量是独立进行求解的，求解过程中采用隐式格式在各个网格点上单独进行求解。在 FLUENT 中，方程用 Gauss-Seidel 法与代数多重网格法（AMG）联合求解。

耦合算法中既可以用显式格式，也可以用隐式格式。在使用隐式格式时，耦合算法以求解方程组（而不是单独求解某个方程）的形式求出流场变量，因此方程求解使用块 Gauss-Seidel 法与 AMG 法联合完成。在使用显式格式时，方程组用 Runge-Kutta 法求解，同时用全逼近存储（Full Approximation Storage，FAS）多重网格法加速计算。

6.2 离散化

FLUENT 采用有限体积法将非线性偏微分方程转变为网格单元上的线性代数方程，然后通过求解线性方程组得出流场的解。网格划分可以将连续的空间划分为相互连接的网格单元。每个网格单元由位于几何中心的控制点和将网格单元包围起来的网格面或线构成。所谓求解流场控制方程，最终目的就是获得所有控制点上流场变量的值。

在有限体积法中，控制方程首先被写成守恒形式。从物理角度看，方程的守恒形式反映的是流场变量在网格单元上的守恒关系，即网格单元内某个流场变量的增量等于各边界面上变量的通量的总和。有限体积法的求解策略就是用边界面或线上的通量计算出控制点上的变量。比如对于密度场的计算，网格单元的控制点上的密度值及其增量代表的是整个网格单元空间上密度的值和增量。从质量守恒的角度来看，流入网格的质量与流出网格的质量应该等于网格内流体质量的增量，因此从质量守恒关系（连续方程）可以得知密度的增量等于边界面或线上密度通量的积分。

FLUENT 中用于计算通量的方法包括一阶迎风格式、指数律格式、二阶迎风格式、QUICK 格式、中心差分格式等形式，本节将分别进行介绍。

6.2.1 一阶迎风格式

“迎风”这个概念是相对于局部法向速度定义的。所谓迎风格式就是用上游变量的值计算本地的变量值。在使用一阶迎风格式时，边界面上的变量值被取为上游单元控制点上

的变量值。

6.2.2 指数律格式

指数律格式认为流场变量在网格单元中呈指数规律分布，在一维问题中这个指数规律如（6-1）式所示，这个公式由求解一维对流扩散方程得出。在对流作用起主导作用时，指数律格式等同于一阶迎风格式；在纯扩散问题中，对流速度接近于零，指数律格式等于线性插值，即网格内任意一点的值可以用网格边界上的值线性插值得到，而在对流和扩散作用相差无几时，需要用（6-1）式计算得出。

$$\frac{\phi(x)-\phi_0}{\phi_L-\phi_0}=\frac{\exp\left(Pe\dfrac{x}{L}\right)-1}{\exp(Pe)-1} \tag{6-1}$$

ϕ_0 表示一维网格起点处的变量值，ϕ_L 表示网格终点处的变量值，Pe 为表征对流作用与扩散作用之比的 Peclet 数。对流作用强时，Pe 增大；反之，则 Pe 减小。

6.2.3 二阶迎风格式

一阶迎风格式和二阶迎风格式都可以看作流场变量在上游网格单元控制点展开后的特例：一阶迎风格式仅保留 Taylor 级数的第一项，因此认为本地单元边界点的值等于上游网格单元控制点上的值，其格式精度为一阶精度。二阶迎风格式则保留了 Taylor 级数的第一项和第二项，因而认为本地边界点的值等于上游网格控制点的值与一个增量[1]的和，因而其精度为二阶精度。

6.2.4 QUICK 格式

QUICK 格式用加权和插值的混合形式给出边界点上的值。QUICK 格式是针对结构网格，即二维问题中的四边形网格和三维问题中的六面体网格提出的，但是在 FLUENT 中，非结构网格计算也可以使用 QUICK 格式选项。在非机构网格计算中，如果选择 QUICK 格式，则非六面体（或四边形）边界点上的值是用二阶迎风格式计算的。在流动方向与网格划分方向一致时 QUICK 格式具有更高的精度。

[1] 这个增量等于流场变量的梯度与从控制点到边界点的矢径的点积。

6.2.5 中心差分格式

在使用 LES 湍流模型时，可以用二阶精度的中心差分格式计算动量方程，并得到精度更高的结果。

以本地网格单元的控制点为基点，对流场变量做 Taylor 级数展开并保留前两项，也可以得出边界点上具有二阶精度的流场变量值。在一般情况下，这样求出的边界点变量值与二阶迎风差分得到的变量值不同，而二者的算术平均值就是流场变量在边界点上的用中心差分格式计算出的值。

6.2.6 亚松弛方法

因为流场的控制方程是非线性的，所以 FLUENT 中采用亚松弛法控制流场变量的增量，即令经过迭代得到的增量略小于实际计算值：

$$\phi = \phi_{old} + \alpha \Delta \phi \tag{6-3}$$

式中 ϕ 为实际计算结果，ϕ_{old} 为前一步的计算结果，α 为松弛因子，$\Delta\phi$ 为计算中得到的增量。

6.3 多重网格法

多重网格法是加速计算的有效手段，其基本原理是利用网格由细到粗，再由粗到细的循环过程，将流场的全局误差由低频误差转换为高频误差，从而可以利用现有格式可以有效降低高频误差的特点加速收敛，提高现有计算格式降低全局误差的速度，进而提高整个计算过程的计算速度。

FLUENT 中采用的 Gauss-Seidal 点隐格式和显式多步求解格式可以快速降低局部误差（高频误差），但是因为这些格式是以控制点为中心的，所以在降低全局误差（低频误差）方面效率很低，用于降低全局误差的计算步数通常比降低局部误差的计算步数高出 2～3 个数量级，因此加速全局误差的降低速度就可以大大加快流场计算的收敛速度。

FLUENT 中采用的多重网格法包括 AMG（Algebraic Multi-Grid，代数多重网格法）法和 FAS（Full-Approximation Storage，全逼近存储法）两种方法。其中 AMG 方法是 FLUENT 中用于多重网格计算的基本算法，既可以与分离求解器配合使用，也可以与耦合求解器配合使用；FAS 方法在计算非线性问题方面优于 AMG 方法，但是只能与耦合显式求解器配合使用，同时因为 FAS 方法在计算过程中需要存储每一层网格，所以消耗的计算资源比较多，建议在计算机配置比较高时采用。

用多重网格法进行计算时，由细网格到粗网格的变化过程被称为约束（Restriction），由粗网格到细网格的变化过程被称为扩张（Prolongation）。在这两个过程中，离散方程的形式会相应发生变化。每遍历一次由细到粗，再由粗到细的过程被称为一个循环，根据循环过程的不同，又可以进一步将循环类型划分为 V 循环、W 循环、Flex（灵活）循环，以及由 V 循环和 W 循环为基本结构组成的 F 循环。V 循环的基本特征是在一个循环过程中只计算一次粗网格流场，形象地说是只有一次“触底”过程，而 W 循环则有二次触底过程，F 循环则由一次 W 循环和紧跟在后面的一次 V 循环组成。Flex 循环是根据实际计算情况灵活调整循环设置的循环类型，即在收敛速度很慢时访问粗网格层以加速收敛，在粗网格层上计算收敛后再返回细网格层进行计算。

多重网格计算中究竟设置多少层网格在理论上没有限制，就大多数情况而言，3~5 个网格层就足够了。粗网格的构建方法对于结构网格和非结构网格是不同的。对于结构网格，只要以细网格为基础，去掉一些网格分隔线就可以达到粗化网格的目的，同时计算中的约束和扩张过程也比较容易通过线性插值的形式完成，比如一个“田”字形的网格可以通过去掉中间的网格分隔线变成“口”字形的粗网格。而非结构网格中粗网格的生成方法比较复杂。在 AMG 方法中，粗网格是通过将流场变量关联较强的相邻网格合并得到的。在 FAS 方法中，粗网格是通过将节点附近的所有网格融合为一个网格得到的。

6.4 使用求解器的基本步骤

使用求解器的基本步骤如下：

（1）选择计算格式。在使用分离求解器时，选择压强插值格式。

（2）在使用分离求解器时，选择压强和速度的耦合方法。

（3）在使用分离求解器时，选择多孔介质速度方法。

（4）选择用于计算导数的梯度选项。

（5）设置亚松弛因子。

（6）在使用耦合求解器时，选择使用 FAS 多重网格法。

（7）对所选用的模型和求解器进行细节设置。

（8）对流场进行初始化。

（9）选择求解过程监视器。

（10）开始迭代计算。

（11）如果计算不收敛，尝试本章最后一节中关于改善收敛性的方法。

在随后的几节中将对上面步骤中的重点内容进行介绍。

6.5 选择离散格式

在计算之前可以为对流项的计算选择离散格式，粘性项则自动采用二阶格式。在缺省情况下，分离求解器用一阶迎风格式计算所有方程的对流项，耦合求解器则用二阶格式计算连续、动量和能量方程等主控方程，仅用一阶格式计算湍流和辐射等辅助方程。在选用分离求解器时可以选择压强插值格式。

6.5.1 计算格式的选择

对流项的计算格式

在流动方向与网格方向基本一致时，可以选择一阶迎风格式。一阶迎风格式适用于结构网格，具有稳定性高，计算速度快的优点，但是在网格方向与流动方向不一致时，产生的数值误差比较大。在非结构网格计算，或流场比较复杂的情况下，很难保证流动方向与网格相一致，此时应该选用二阶格式。

二阶格式的计算精度高于一阶格式，但是相对而言，其计算时间比较长，收敛性也相对较差，因此在实际计算中进行格式选择时，需要兼顾精度、收敛性和系统资源等方面的要求。比如可以在计算开始时先用一阶格式进行计算以获得一个相对粗糙的解，在计算收敛后再用二阶格式完成计算以提高解的精度。这样既避免了二阶格式收敛性差、计算时间长的问题，又避免了一阶格式在复杂流场计算中数值误差大的问题。除此之外，在采用二阶格式无法收敛的情况下，也可以考虑改用一阶格式进行计算。

在用结构网格计算旋转流动问题时，QUICK 格式可以提供更高的计算精度，但是在其他情况下 QUICK 格式的精度与二阶格式相当，并没有很大的改进。计算中还可以选用指数律格式，其精度与一阶格式基本相同。中心差分仅在使用 LES 湍流模型时可以使用，并且应该在网格足够密集、局部 Peclet 数小于 1 的情况下使用。

压强插值格式的选择

在大多数情况下，可以使用 FLUENT 缺省设置的“标准”格式，对于下列特殊情况可以分别选择相应的格式：

- 在彻体力对流场有很大影响的情况下，应该选择彻体力加权（body-force-weighted）格式。
- 在流场中有涡量很大的集中涡、高雷诺数自然对流、高速旋转流、多孔介质，以及流线曲率很大时，应该选择 PRESTO!格式。
- 对于可压流，应该使用二阶格式。
- 二阶格式不能用于多孔介质计算和多相流计算中的混合物模型及 VOF 模型。在其他情况下，为了提高精度可以选用二阶格式。

密度插值格式

在用分离算法计算单相可压流时，有三种密度插值格式可供选择，即一阶迎风格式、二阶格式和 QUICK 格式。一阶迎风格式具有良好的稳定性，但是在计算带激波的可压流时，会对激波解产生“抹平”作用，因此应该选用二阶格式或 QUICK 格式。在用四边形网格、六面体网格或混合网格计算带激波的流动时，最好使用 QUICK 格式计算所有变量。需要注意的是，在计算可压多项流时，只能用一阶迎风格式计算可压缩相的流动。

6.5.2 软件中的实际操作过程

在 FLUENT 软件中选择计算格式是在 Solution Controls（求解过程控制）面板中完成的，如图 6-3 所示。进入这个面板的菜单操作步骤如下：

Solve->Controls->Solution...

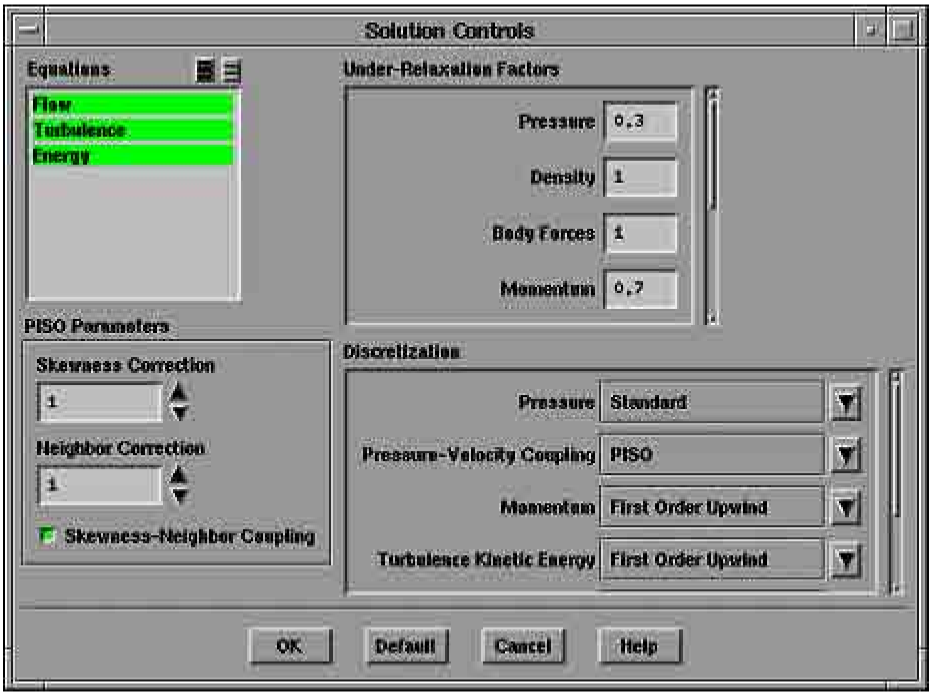


图 6-3 求解过程控制面板

在 Discretization（离散）组中，可以在使用分离求解器时，定义动量、能量、湍流动能等项目，并为这些项目选择的一阶迎风格式、二阶迎风格式、指数律格式、QUICK 格式和中心差分格式（在 LES 湍流模式计算中），也可以在使用耦合求解器时，定义湍流动能、湍流耗散率等项目，并为这些项目选择一阶迎风格式、二阶迎风格式。

需要注意的是，图 6-3 是选择分离算法时 Solution Controls（求解过程控制）面板的外

观，采用耦合算法时这个面板的外观有所不同。

在使用分离算法时，可以在 Pressure（压强）旁边的列表中选择压强插值格式，其中包括标准格式、线性格式、二阶格式、彻体力加权格式和 PRESTO!格式。在计算可压流时，需要在 Density（密度）旁边的列表中选择密度插值格式，包括一阶迎风、二阶迎风和 QUICK 格式。

按 Default（缺省）按钮可以将所有设置恢复为缺省设置。

6.6 选择压强－速度关联算法

在使用分离求解器时，可以选择三种压强－速度的关联形式，即 SIMPLE、SIMPLEC 和 PISO。SIMPLE 和 SIMPLEC 通常用于定常流计算，PISO 用于非定常计算，但是在网格畸变很大时也可以使用 PISO 格式。

6.6.1 格式选择

FLUENT 缺省设定的格式为 SIMPLE 格式，但是因为 SIMPLEC 稳定性较好，在计算中可以将亚松弛因子适当放大，所以在很多情况下可以考虑选用 SIMPLEC。特别是在层流计算时，如果没有在计算中使用辐射模型等辅助方程，用 SIMPLEC 可以大大加速计算速度。在复杂流动计算中，二者收敛速度相差不多。

PISO 格式通常被用于非定常计算，但是它也可以用于定常计算。PISO 格式允许使用较大的时间步长进行计算，因而在允许使用大时间步长的计算中可以缩短计算时间。但是在类似于大涡模拟（LES）这类网格划分较密集，因而时间步长很小的计算中，采用 PISO 格式计算则会大大延长计算时间。另外在定常问题的计算中，PISO 格式与 SIMPLE 和 SIMPLEC 格式相比并无速度优势。

PISO 格式的另一个优势是可以处理网格畸变较大的问题。如果在 PISO 格式中使用邻近修正[2]（neighbor correction），可以将亚松弛因子设为 1.0 或接近于 1.0 的值。而在使用畸变修正（skewness correction）时，则应该将动量和压强的亚松弛因子之和设为 1.0，比如将压强的亚松弛因子设为 0.3，将动量的亚松弛因子设为 0.7。如果同时采用两种修正形式，则应将所有松弛因子设为 1.0 或接近于 1.0 的值。

在大多数情况下都不必修改缺省设置，而在有严重网格畸变时，可以解除邻近修正和畸变修正之间的关联关系。

[2]即对动量进行修正

6.6.2 在软件中设置格式

在 Solution Controls（求解过程控制）面板中（见图 6-3），在 Discretization（离散）组下面的压强速度耦合列表中选择 SIMPLE、SIMPLEC 或 PISO 格式即可。

如果选择 PISO，则面板会自动展开出现 PISO 参数设置栏。在缺省情况下，畸变修正（Skewness Correction）和邻近修正（Neighbor Correction）的值都是 1。如果仅使用畸变修正，则将邻近修正的值设为 0，反之，则将畸变修正的值设为 0。缺省情况下，Skewness-Neighbor Coupling（畸变邻近关联）选项是被选中的，可以根据情况取消二者的关联关系。

6.7 设置亚松弛因子

如前所述，FLUENT 中各流场变量的迭代都由亚松弛因子控制，因此计算的稳定性与亚松弛因子紧密相关。在大多数情况下，可以不必修改亚松弛因子的缺省设置，因为这些缺省值是根据各种算法的特点优化得出的。在某些复杂流动情况下，缺省设置不能满足稳定性要求，计算过程中可能出现振荡、发散等情况，此时需要适当减小亚松弛因子的值，以保证计算收敛。

在实际计算中可以用缺省设置先计算 4、5 步，如果发现残差曲线向上发展，则中断计算，适当调整亚松弛因子后再继续计算。在修改计算控制参数前，应该先保存当前计算结果。调整参数后，计算需要经过几步调整才能适应新的参数。一般而言，增加亚松弛因子将使残差增加，但是如果格式是稳定的，增加的残差仍然会逐渐降低。如果改变参数后，残差增加了几个量级，就可以考虑中断计算，并重新调入保存过的结果，再做新的调整。

在计算发散时，可以考虑将压强、动量、湍流动能和湍流耗散率的亚松弛因子的缺省值分别降低为 0.2、0.5、0.5、0.5。在计算格式为 SIMPLEC 时，通常没有必要降低亚松弛因子。

亚松弛因子是在 Solution Controls（求解过程控制）面板中 Under-Relaxation Factors（亚松弛因子）旁的输入栏中设定的。点击 Default（缺省）按钮可以恢复缺省设置。

6.8 改变库朗数

FLUENT 中的时间步长是由库朗数定义的，而库朗数是由线性稳定性理论定义的一个范围，在这个范围内计算格式是稳定的。给定一个库朗数，就可以相应地得到一个时间步长。库朗数越大，时间步长就越长，计算收敛速度就越快，因此在计算中库朗数都在允许的范围内尽量取最大值。

6.8.1 耦合显式格式中的库朗数

在显式格式和隐式格式中库朗数的取值范围差别很大，在显式格式中库朗数的取值范围很小，隐式格式则宽松得多。

在 FLUENT 使用的多步格式中，可以认为库朗数在小于 2.5 的范围内是稳定的。由于控制方程是非线性的，所以库朗数的取值一般达不到线性稳定性分析得到的极限值。在耦合算法的显式格式中，系统设定的缺省值为 1.0，在某些二维问题中可以适当放大这个值，但是不要超过 2.0。

如果计算模型的设定是正确的，并且进行了初始化，而计算中却发现残差快速上升，通常说明库朗数的值可能需要降低。另外，在计算的开始阶段，因为初始流场相对粗糙，所以可以适当降低库朗数，比如降低到 0.1－0.5 左右，然后在计算相对稳定后再适当调高库朗数。

6.8.2 耦合隐式格式中的库朗数

在线性稳定性理论表明 FLUENT 中使用 Gauss-Seidel 格式是无条件稳定的，但是由于控制方程非线性的影响，实际上库朗数在这种情况下也不可能取为无限大。在隐式格式中，库朗数的缺省值为 5.0，在很多情况下可以将缺省值改为 10、20、100 甚至更高，具体取多高主要取决于问题的复杂程度。与显式格式一样，在计算开始的时候可以把库朗数取得小一些，而在经过几个迭代步后可以将库朗数再调高。

AMG 求解器可以在多重网格的计算过程中发现计算是否已经发散。如果计算开始发散，AMG 求解器会自动降低库朗数并重新进行迭代，同时屏幕上会显示相关信息。在迭代成功后，库朗数会恢复原来的值，迭代计算则继续进行下去。如果迭代不成功，计算就会中断。

6.8.3 库朗数在 FLUENT 中的设定

库朗数是在 Solution Controls（求解过程控制）面板中设定的。在 Solver Parameters（求解器参数）组中的 Courant Number（库朗数）旁边的栏中输入其值就可以了。如果选择的求解器是显式格式，则系统会自动将库朗数设为缺省值 1。如果选择的求解器是隐式格式，则系统设定的缺省值为 5。

6.9 引入 FAS 多重网格

如前面章节所述，FAS 多重网格是耦合算法显式格式的一个可选项。因为几乎所有耦

合显式格式计算都可以在使用 FAS 多重网格法时得到加速，所以在计算中应该尽量采用 FAS 多重网格法。采用多重网格法唯一需要设定的参数是网格的级数，只要将这个参数设置为不等于零的数，系统就自动在计算中启用 FAS 多重网格法。

需要注意的是，AMG 代数多重网格法在缺省设置中是永远被使用的方法。另外，FAS 多重网格法不能与显式时间步长配合使用，因为粗网格计算将打乱细网格计算中对时间步长的计算。

如前所述，FAS 多重网格法在求解过程中将粗网格的解传递回细网格，因此可以加快流场变量信息的传播速度，并加快计算收敛进程。控制多重网格求解过程的基本办法是设置粗网格的级数。在多重网格法中，虽然粗网格的级数越多，收敛就越快，但是也不能无限制增加粗网格级数，因为级数越多，占用系统资源也越多。同时粗网格是由临近的细网格结合而成的，理论上存在一个极限，即在某个级数上可能其次级“细”网格仅有一个网格单元，此时再增加网格级数已无意义，FLUENT 在这种情况下会假定高级网格等同于其次级网格，而不再增加网格级数。

在 FLUENT 软件中可以在两个地方设置网格级数，即 Solution Controls（求解控制）面板和 Multigrid Controls（多重网格控制）面板，Multigrid Controls（多重网格控制）面板可以用下面的菜单操作进入：

Solve->Controls->Multigrid...

改变粗网格级数，系统就会自动更新内部设置，启用 FAS 多重网格法。

6.10 设置求解极限

流场变量在计算过程中的最大、最小值可以在 Solution Limits（求解极限）面板中设定，菜单操作次序为：

Solve->Controls->Limits...

设置解变量极限是为了避免在计算中出现非物理解，比如密度或温度变成负值，或者大得远远超过真实值。FLUENT 中同时可以对温度的变化率极限进行设置，这样可以避免因为温度变化过于剧烈而导致温度出现负值，温度变化率的缺省设置是 0.2，即温度变化率不能超过 20％。

在计算之前可以对缺省设定的解变量极限进行修改，比如温度的缺省设置是 5000K，但是在一些高温问题的计算中，可以将这个值修改为更高的值。另外，如果计算过程中解变量超过极限值，系统会在屏幕上发出提示信息，提示在哪个计算区域、有多少网格单元的解变量超过极限。对湍流变量的限制是为了防止湍流变量过大，对流场造成过大的、非物理的耗散作用。

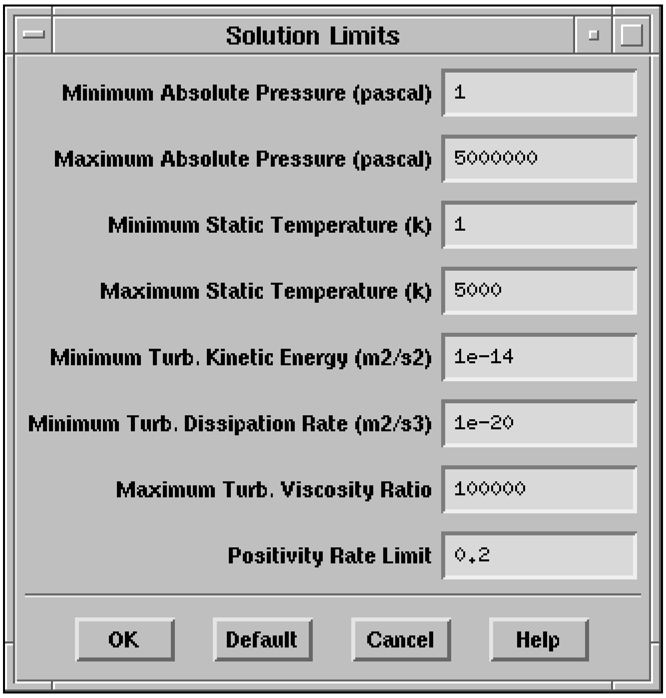


图 6-4 解变量极限面板

6.11 初始化

在开始进行计算之前，必须为流场设定一个初始值。设定初始值的过程被称为“初始化”。如果把每步迭代得到的流场解按次序排列成一个数列，则初始值就是这个数列中的第一个数，而达到收敛条件的解则是最后一个数。显然如果初始值比较靠近最后的收敛解，则会加快计算过程，反之则会增加迭代步数，使计算过程加长，更严重的是如果初始值给的不好，有可能得不到收敛解。

在 FLUENT 中初始化得方法有两种，即

（1）全局初始化，即对全部网格单元上的流场变量进行初始值设置。

（2）对流场进行局部修补，即在局部网格上对流场变量进行修改。

在进行局部修补之前，应该先进行全局初始化。

6.11.1 全局初始化

计算前对流场进行的全局初始化是在 Solution Initialization（初始化）面板中进行设置的。用下列次序操作可以进入这个面板：

Solve->Initialize->Initialize...

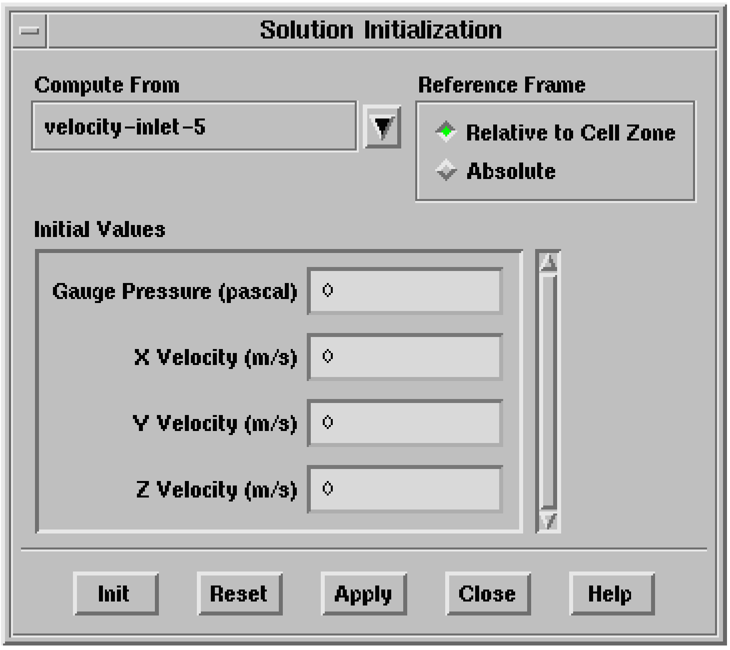


图 6-5 初始化面板

初始化的步骤如下：

（1）设定初始值：

- 如果想用某个区域上设定的初始值进行全局初始化，应该先在 Compute From（计算起始位置）列表中选择需要定义初始值的区域名，再在 Initial Values（初始值）中给定各变量的值，则所有流场区域的变量的值都会根据给定区域的初始值完成初始化过程。
- 如果用平均值的办法对流场进行初始化，则在 Compute From 列表中选择 all-zones（所有区域），则 FLUENT 将根据边界上设定的值计算出初始值，完成对流场的初始化。
- 如果希望对某个变量的值做出改变，可以直接在相应的栏目中输入新的变量值。

（2）如果计算中使用了动网格，可以通过选择“Absolute（绝对速度）”或“Relative to Cell Zone（相对于网格区域）”来决定，设定的初始值为绝对速度值，还是相对速度。缺省设置为相对速度。

（3）在检查过所有初始值的设定后，可以点击 Init（初始化）按钮开始流场的初始化。如果初始化是在计算过程中重新开始的，则必须用 OK 键确认用新的初始值覆盖计算值。

初始化面板下面几个按钮的含义为：

- Init（初始化）按钮，保存初始值设置，并进行初始化计算。
- Apply（应用）按钮，仅进行初始值保存操作。
- Reset（重置）按钮，如果初始化过程有错误，比如初始值有错误，或者使用了错

误的区域作为开始区域，则可以按此键将初始值恢复为缺省值。

6.11.2 对初始值进行局部修补

在完成全局初始化后，可能会对某些变量的值进行局部修补。局部修补在修补（Patch）面板中进行设定。

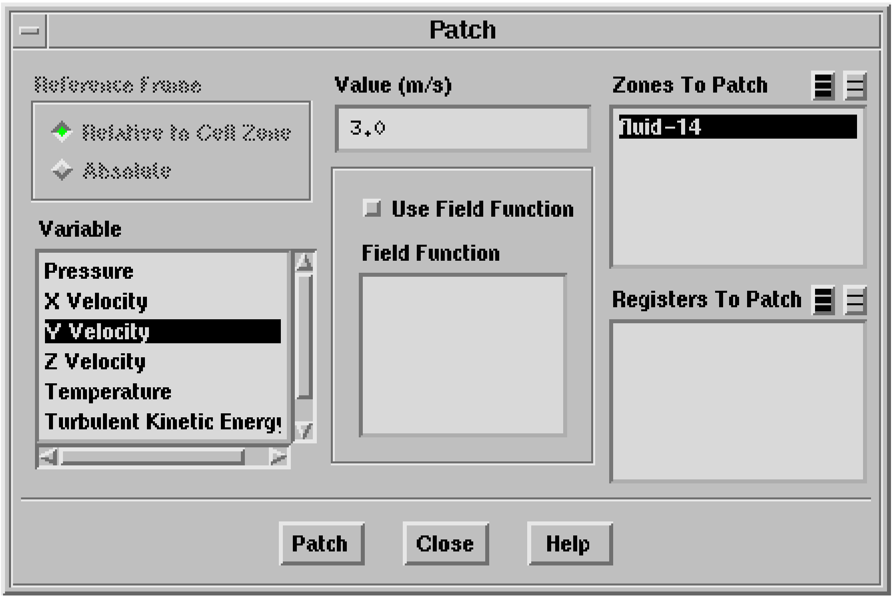


图 6-6 修补面板

局部修补的步骤如下：

（1）在 Variable（变量）列表中选择需要修补的变量名。

（2）在 Zones To Patch（需要修补的区域）或 Registers To Patch（需要修补的标记区）中选择需要修补变量所在的区域。

（3）如果需要将变量的值修补为常数，则直接在输入栏中输入变量的值。如果需要用一个预先设定的函数定义变量，可以在 Use Field Function（使用场变量函数）中的 Field Function（场函数）列表中选择合适的场函数。

（4）如果需要修补的变量为速度，则除了定义速度的大小，还要定义速度为绝对速度，还是相对速度。

（5）点击 Patch（修补）按钮更新流场数据。

局部修补通常是针对某个流场区域进行的，而用标记区进行局部修补则可以对某个流场区域中的一部分网格上的变量值进行修补。标记区可以用网格的物理坐标、网格的体积特征、变量的梯度或其他参数进行标记。在创建了标记区后，就可以对标记区上的初始值进行局部修补操作了。

用 Custom Field Function Calculator（编制场函数算子）面板可以编制自己的场函数，然后用场函数来反映物理量在流场中的变化过程。

因为局部修补不影响流场的其他变量，所以也可以在计算过程中，用局部修补的方法改变某些变量的值，对计算过程进行人为干预。

6.12 流场求解

6.12.1 定常问题计算

定常状态的计算在 Iterate（迭代）面板中开始，迭代面板的开启过程如下：

Solve->Iterate...

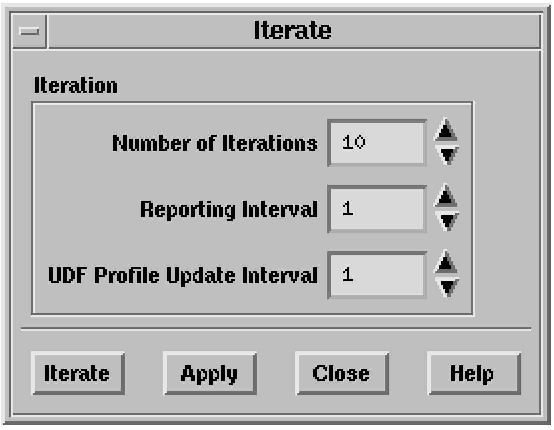


图 6-7 迭代面板

迭代面板中第一个输入栏为 Number of Iterations（迭代步数），在这里填入计算需要迭代的步数。第二栏为 Reporting Interval（报告间隔），即每隔多少步显示一次求解信息，缺省设置为 1。如果计算中使用了 UDF 函数，则可以用第三行中的输入栏决定每隔多少步输出一次 UDF 函数的更新信息。

设置完毕后，点击 Iterate（迭代）按钮就可以开始计算了。在计算开始后，会弹出一个工作窗口提示迭代正在进行，如果想中断计算，可以点击窗口中的 Cancel（取消）按钮就可以中断计算。如果想继续计算，则再次点击 Iterate 按钮就可以了。如果对计算结果不满意，希望重新开始计算的话，则要重新初始化一次。

6.12.2 时间相关问题的计算

用户输入

FLUENT 可以模拟的时间相关问题包括旋涡卷动、可压缩振荡、瞬态热传导、瞬态化学反应与混合流动等问题，比如圆柱后面卡门涡街中旋涡的生成和脱落过程等等。

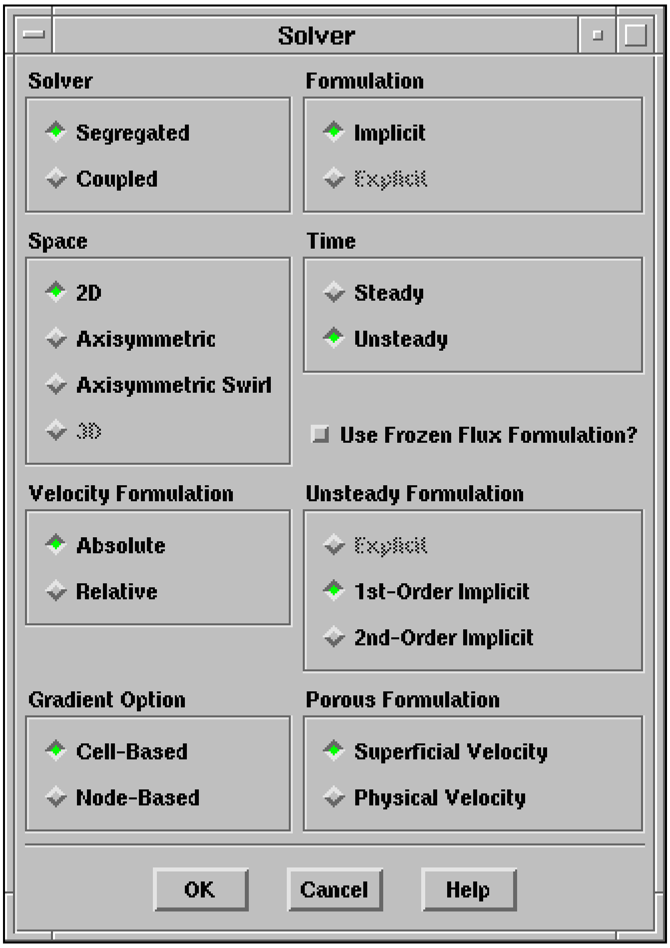


图 6-8 求解器面板

求解时间相关问题时，求解器的设置步骤如下：

（1）求解器面板的启动步骤为：

Define->Models->Solver...

首先选中 Unsteady（非定常）选项，然后再在 Unsteady Formulation（非定常算法）中选择所需要的算法。一般地说，1st-Order Implicit（一阶隐式格式）适用于大多数情况。如果要提高精度，可以使用 2nd-Order Implicit（二阶隐式格式）。在选择耦合显式求解器时，

可以使用 Explicit（显式）格式。显式格式主要用于激波等波动解的捕捉问题。

在使用分离求解器时，还可以选择 Use Frozen Flux Formulation（使用冻结通量格式）选项，但是这个选项只能在用分离求解器计算单相流问题时使用，同时计算中还不能使用动网格技术。

（2）定义相关物理模型和边界条件。

（3）如果选用了分离求解器，则应该在压强－速度关联方法中选择 PISO 格式。在用 LES 湍流模型计算湍流问题时，不应该选择 PISO 格式，那样会增加系统资源消耗。

（4）如果使用的是显式非定常计算或适应性时间推进法，应该在每步迭代中打开当前时间或当前时间步长的 print 选项。这项操作在 Statistic Monitors（统计数据监视器）面板中完成，通过下列菜单操作可以进入这个面板：

Solve->Monitors->Statistic...

在 Statistics（统计数据）列表中选择 time（时间）或 delta_time（时间增量），并打开 Print 选项。当 FLUENT 在控制台窗口打印残差信息时，就会出现当前时间或当前时间步长。

（5）可以用 Force Monitors（受力监视器）或 Surface Monitors（表面监视器）面板监视随时间变化的受力系数的值或平均值、质量平均值、积分或场变量通量，或者任意一个函数在表面上随时间变化的情况。

（6）用 Solution Initialization（求解初始化）面板设定初始条件：

Solve->Initialize->Initialize...

还可以读入一个定常状态的数据文件来设定初始条件：

File->Read->Data...

（7）用自动保存功能定义文件名和在迭代过程中自动保存算例文件和数据文件的频率：

File->Write->Autosave...

还可以用 Execute Commands（执行命令）面板自动执行系统命令。

（8）如果想创建解的动画过程，可以使用 Solution Animation（求解动画）面板设置动画显示参数。

（9）监视时间数据的步骤如下：

1）在 Iterate（迭代）面板中打开 Data Sampling for Time Statistics（时间数据采样）选项：

Solve->Iterate...

打开这个选项可以同时显式平均值和均方根值。

2）初始化流场统计数据：

Solve->Initialize->Reset Statistics

用这个菜单还可以在收集了一些数据后重置流场数据。比如在将时间采样选项打开的情况下先计算 10 步，检查计算结果，然后再计算 10 步，则时间统计数据中将包括前面 10 步的数据，除非重新对流场统计数据进行初始化。

（10）指定时间相关解的参数并开始计算，对于定常、非定常计算的步骤如下：

1）如果选择一阶或二阶精度隐式计算，步骤如下：

□ 在 Iterate（迭代）面板中定义时间相关解的参数：

Solve->Iterate...



图 6-8 迭代面板

隐式非定常计算公式参数包括每个时间步的 Max Iterations per Time Step（最大迭代次数）和 Time Step Size（时间步长的大小）。虽然从线性稳定性理论分析，全隐式格式是无条件稳定的，但是如前所述，在非线性系统中，时间步长还是要受到一定的限制的。通常在计算开始时，时间步长取得较小，而在后续的计算中则可以逐步增加时间步长。在缺省设置中，时间步长是固定的，如果需要让时间步长随计算进程发生变化，可以选择 Adaptive Time Stepping（适应性时间推进）法中的 Adaptive（适应性步长）选项。

□ 定义计算所需要的 Number of Time Steps（时间步数量），然后点击 Iterate 按钮开始计算。

2）显式非定常计算的步骤如下：

□ 在 Solution Controls（求解过程控制）面板中保留 Solver Parameters（求解器参数）缺省设置：

Solve->Controls->Solution...

如果对 Solver Parameters 进行过修改，可以点击 Default（缺省）按钮恢复缺省设置。

- 输入所需要的 Number of Iterations（迭代步数），点击 Iterate 按钮开始计算：

Solve->Iterate...

（11）保存数据文件和算例文件，以便将来可以继续进行计算：

File->Write->Data...

适应性时间推进

如前所述，时间步长在计算过程中是可以随着截断误差的变化而变化的。在截断误差小于某个给定值时，时间步长将增加。在截断误差大于某个给定值时，时间步长将减小。使用预报校正格式可以计算出截断误差的值，并根据这个值相应调整时间步长。在每个时间步中，预报步可以用耗时较少的显式格式得到一个流场解，这个流场解可以作为校正步的初始条件用于校正步的计算。校正步通常用非线性迭代过程进行计算，校正步和迭代步中得到的两个解的差可以作为截断误差，将这个值与所需要的精度进行对比后，FLUENT 就可以相应增加或减小时间步长。

适应性时间推进的控制参数在 Iterate（迭代）面板中设置，包括：

- Truncation Error Tolerance（截断误差容限），即与截断误差进行对比的判据。增加这个参数的值，则时间步长将增加而计算精度将下降；减小这个值，则时间步长将减小而计算精度将上升。系统设定的缺省值为 0.01。
- Ending Time（结束时间）。因为在适应性时间推进算法中，时间步长是变化的，所以需要事先设定一个结束时间，在累积时间达到结束时间时计算自动结束。
- Minimum/Maximum Time Step Size（最小和最大时间步长），即时间步长的上下限。
- Minimum/Maximum Step Change Factor（最小和最大步长改变因子）是时间步长变化的限制因子，采用这个参数主要是为了限制时间步长发生剧烈变化。如果步长改变因子用 f 表示，并且将其定义为给定截断误差与截断误差容限的比值，则时间步长 Δt_n 用下述方式进行计算：

  在 $1 < f < f_{\max}$ 时，Δt_n 将增加以满足误差容限的要求。

  在 $1 < f_{\max} < f$ 时，Δt_n 将增加，但其最大值为 $f_{\max}\Delta t_{n-1}$。

  在 $f_{\min} < f < 1$ 时，Δt_n 保持不变。

  在 $f < f_{\min} < 1$ 时，Δt_n 将下降。
- Number of Fixed Time Steps（固定时间步的数量），即在时间步长发生变化之前的迭代步数。

6.13 监视计算收敛过程

在计算过程中可以动态监视残差、统计数据、受力值、面积分和体积分等与计算相关的信息，并可以在屏幕上或其他输出设备上输出这些信息。

6.13.1 监视残差

在每个迭代步结束时，都会对计算守恒变量的残差进行计算，计算的结果可以显示在窗口中，并保存在数据文件中，以便随时观察计算的收敛史。从理论上讲，在收敛过程中残差应该无限减小，其极限为 0，但是在实际的计算中，单精度计算的残差最大可以减小 6 个量级，而双精度的残差最大可以减小 12 个量级。

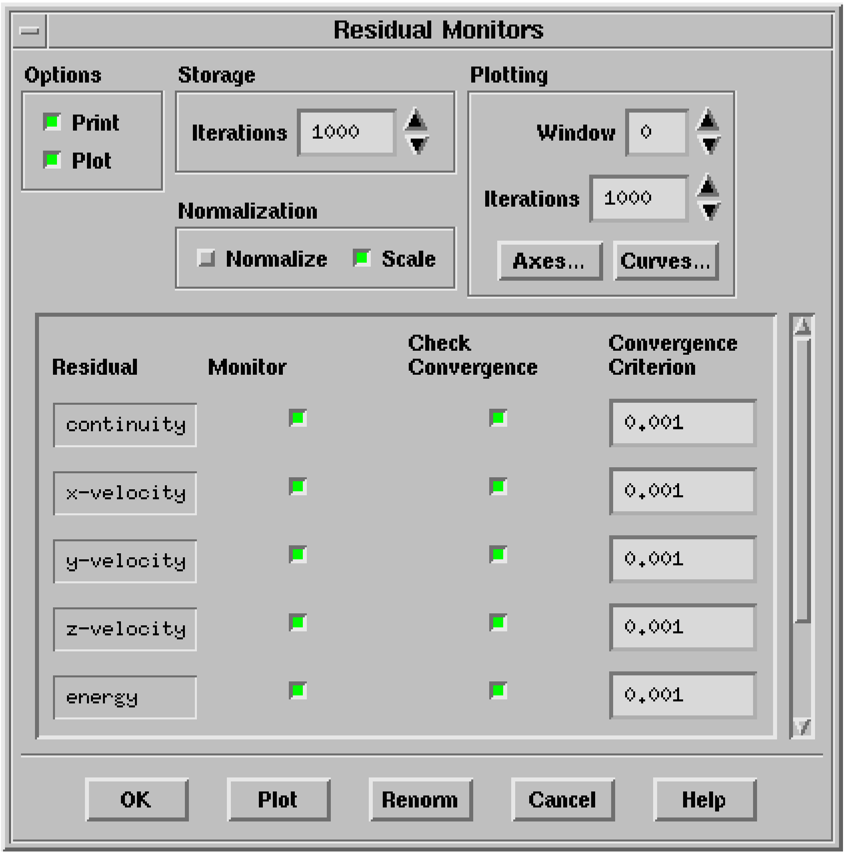


图 6－9 残差监视器面板

分离算法中的残差是守恒方程两端数值的差值在所有网格点上的和，耦合算法中的残差则是各变量的时间变化率在所有网格点上的均方根。在缺省情况下，系统显示的残差是

缩尺残差即残差相对于某个值的比例，其计算方法因变量不同、算法不同而有所不同。而残差的正则化是将残差值除以最大残差值的变换过程。

残差显示设置在 Residual Monitor（残差监视器）面板中进行，如图 6－9 所示，启动残差监视器的菜单操作过程为：

Solve->Monitors->Residual...

或

Plot->Residuals...

在面板的下部可以选择需要监视的变量，并针对各变量设置收敛判据，选择检查该变量是否满足收敛判据等。在面板上方可以选择是否在控制台窗口中以文本方式输出残差的数值（Print 选项），是否绘制残差曲线（Plot 选项），并可以选择保存几个迭代步上的残差值（Iterate 选项），是否对残差进行正则化处理（Normalize 选项），是否进行缩尺处理（Scale 选项），以及残差的窗口显示风格（Plotting 组），包括显示窗口（Window 选项）、显示迭代点数（Iterations）和显示线型、字体等（Axes、Curves）。

6.13.2 监视统计数据

在计算过程中，可以监视周期流动的压强梯度和温度比、非定常流动所用时间、适应性时间推进过程中的时间步长等参数。

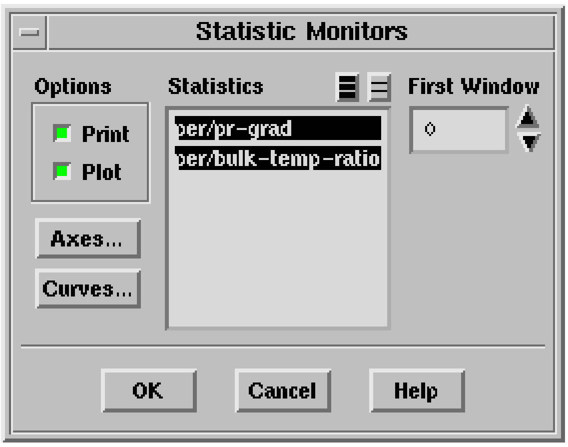


图 6－10 统计数据监视器

统计数据监视器的启动过程如下：

Solve->Monitors->Statistic...

设置监视器的步骤如下：

（1）指定输出类型，即指定是否使用 Print 方式，或 Plot 方式。

（2）在 Statistics 列表中选择需要监视的变量。

（3）如果选择了 Plot（绘图）方式，则可以用 Axes..（坐标轴）面板和 Curves...（曲

线）面板对相关参数进行设置，比如设置显示线型、字体、颜色等等进行设置。

6.13.3 力和力矩监视器

在每次迭代结束后，可以通过计算得到流场中物体所承受的来自流体的力和力矩系数。力和力矩系数也可以通过文本方式（Print 选项）或图形方式（Plot 选项）在屏幕上显示。

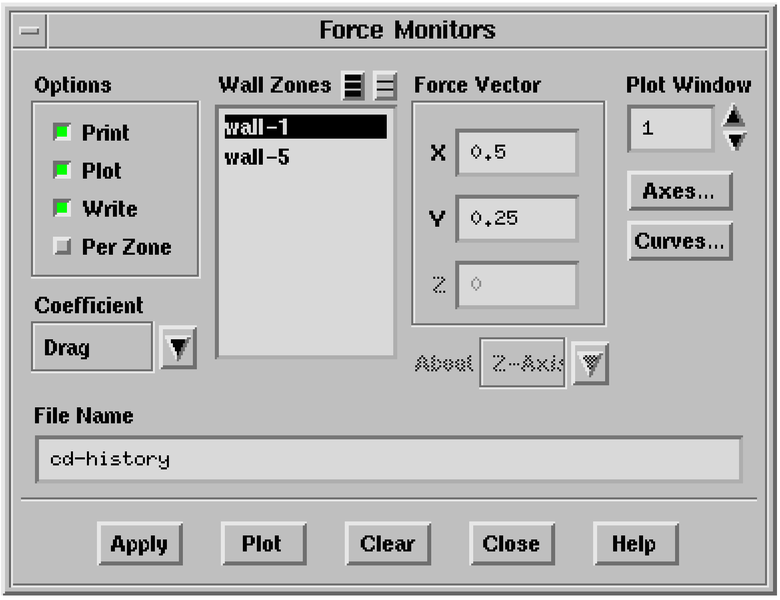


图 6－11 受力监视器

在很多情况下，计算关心的中心问题是物体在流场中受到的力和力矩大小及分布等情况，比如在计算飞机绕流时，最关心的就是飞机行受到的气动力和力矩。在很多情况下，虽然残差仅仅收敛了三个量级，气动力已经收敛。在这种情况下，可以结束计算以节省计算时间，因此对气动力进行监视就很必要。

力和力矩的监视器面板，如图 6－11 所示，通过下列菜单操作被打开：

Solve->Monitors->Force...

相关设置如下：

（1）指定输出类型，即选择以文本形式（Print）输出、图形形式（Plot）输出，还是文件形式（Write）输出。

（2）如果需要对作用在某个壁面上的力和力矩进行监视，可以打开 Per Zone（分区）选项。

（3）在 Coefficient（系数）列表中选择 Drag（阻力）、Lift（升力）或 Moment（力矩），以显示相应的系数。

（4）在 Wall Zones（壁面区）列表中选择壁面名称。

（5）如果选择显示阻力或升力，则在 Force Vector（力矢量）中输入力矢量的 X、Y、Z 分量。如果选择显示力矩，则在 Moment Center（力矩中心）中输入力矩中心的直角坐标值，然后在 About（转动轴方向）列表中选择力矩矢量的方向，即 X-Axis、Y-Axis 或 Z-Axis。

（6）点击 Apply（应用）按钮完成设置。如果需要设置其他参数，则重复上述过程。

6.13.4 监视表面积分

在每次迭代结束后，还可以在某个面上对特定的流场变量进行积分，并以文本、图形和文件形式输出积分结果。比如，在以计算压强为目的的计算中，可以在某个面上监视压强的变化过程。

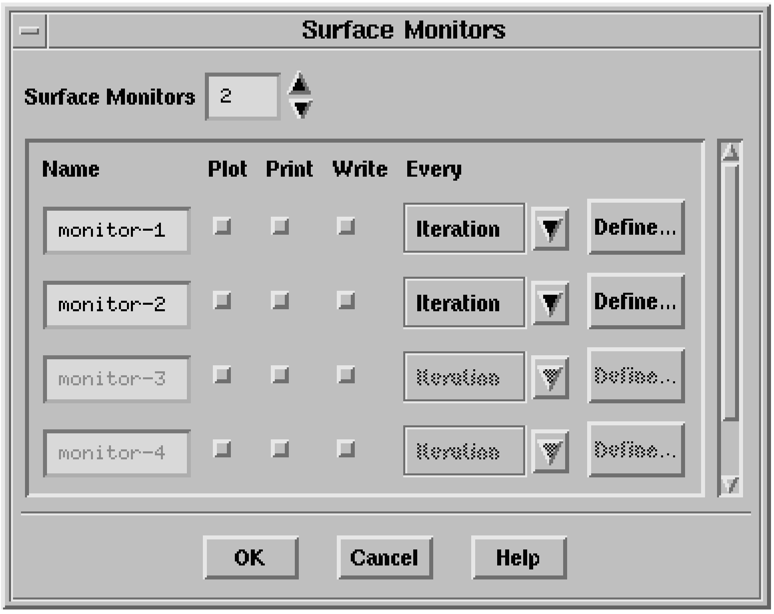


图 6－12 表面监视器

在计算过程中需要监视几个表面上的变量，是在 Surface Monitors（表面监视器）面板中决定的。表面监视器面板的启动操作如下：

Solve->Monitors->Surfaces...

表面监视器的操作过程如下：

（1）在 Surfaces Monitors 旁边的输入栏中填入准备开启的监视器数量，也就是需要监视变量的数量。

（2）在 Name（名称）下面的栏目中加入各监视器的名称。在 Plot、Print、Write 下面选择变量的输出方式，即选择图形方式、文本方式和文件方式。

（3）在 Every 下面选择显示变量的频率，可以选择每个 Iteration（迭代步）、每个 Time Step（时间步）更新一次监视器窗口。

（4）点击 Define（定义）按钮进入 Define Surface Monitors（定义表面监视器）面板，如图 6-13 所示。

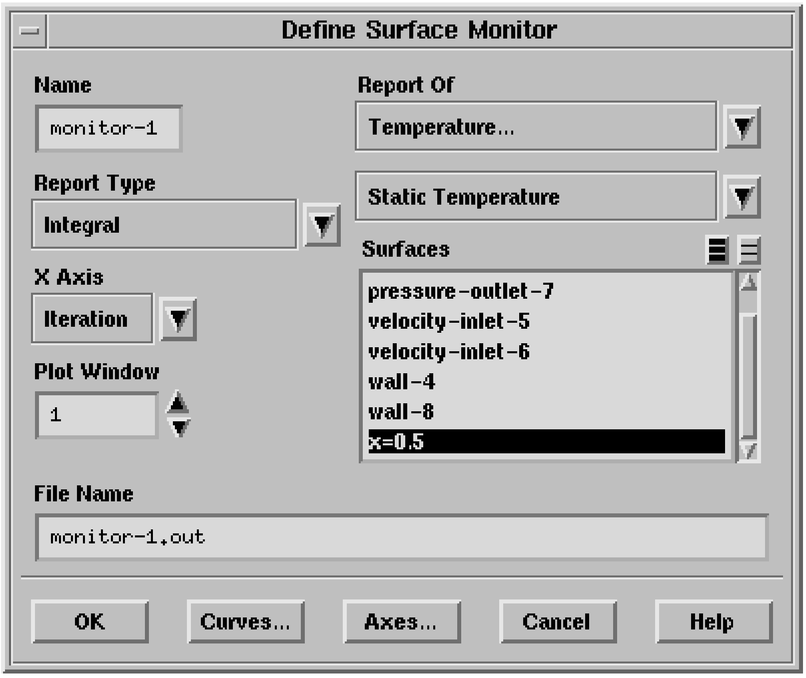


图 6－13 定义表面监视器面板

（6）在 Define Surface Monitors 面板中可以顺序定义监视器名称、报告类型、X 轴变量、显示窗口编号、变量名、积分表面、输出文件名等等，还可以定义图形显示的字体、曲线线型等等。

比如，如果想显示 x=0.5 表面上的静温的积分值，可以先给定监视器的名称，然后在 Report Type 中选择 Integral，在 X Axis 中选择 Iteration，即在显示窗口中 X 轴代表迭代步，在 Report of 中选择 Temperature（温度），在下面的列表中选择 Static Temperature（静温），在 Surfaces 列表中选择 x=0.5，在 File Name 中指定输出文件名，然后点击 OK 即定义完毕。如果对显示效果不满意，可以点击下面的 Curves...按钮和 Axes...按钮对坐标轴及显示曲线等进行详细定义，在大多数情况下没有这个必要。

6.13.5 体积分监视器

与监视面积分类似，在计算过程中还可以监视流场变量的体积分。在体积分监视器中，实际上可以监视的参数还包括流场变量的质量积分、质量平均等等。体积分主要用来监视某个体域内流场变量的变化情况，通过监视变量的体积分可以对求解过程是否收敛做出判断。在使用适应性网格技术的时候，体积分也可以用来判断解是否与网格有关，即在网格

变化的过程中，如果解不随网格变化而变化，则证明计算得到的解与网格无关。

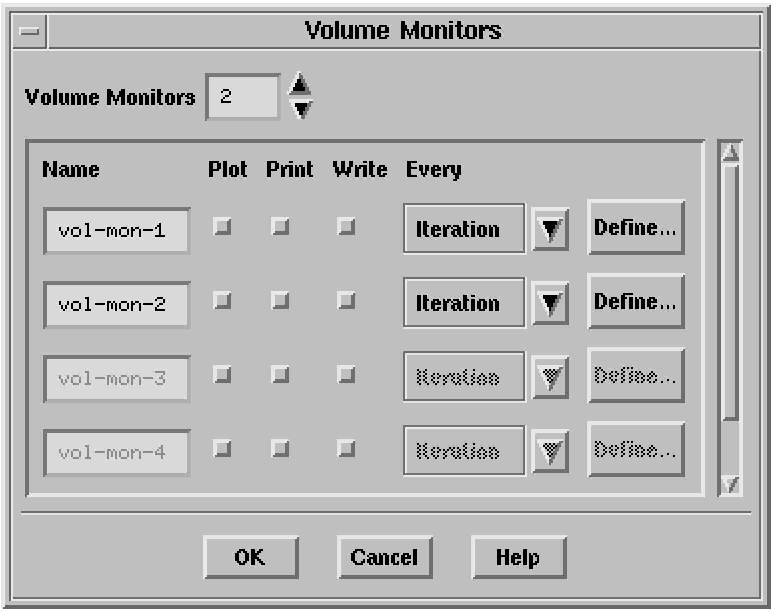


图 6-14 体积分监视器面板

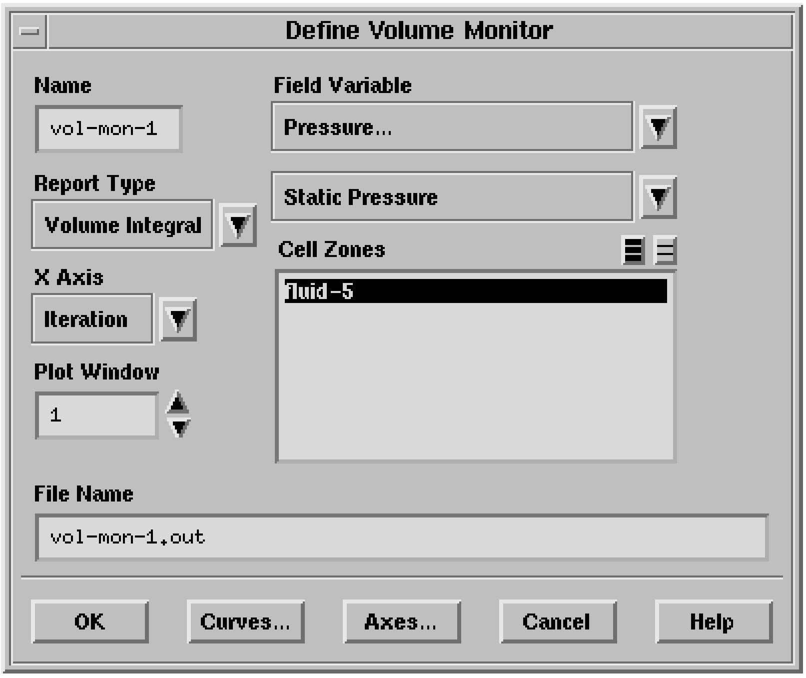


图 6-15 定义体积分监视器面板

体积分监视器在 Volume Monitors（体积分监视器）面板中做总体设置，在 Define Volume

Monitors（定义体积分）面板中做详细设置。体积分监视器的启动过程为：

Solve->Monitors->Volume...

体积分监视器面板与面积分监视器面板是十分相似的，其设置过程也类似，即可以在这里设定窗口名称、窗口数量、显示方式等选项，并通过点击 Define（定义）按钮进入 Define Volume Monitors（定义体积分监视器）面板，如图 6-15 所示。在这个面板中可以指定准备做积分的体域、流场变量、报告类型等等，与面积分定义过程相类似，这里不再重复。

6.14 用动画显示解

在 FLUENT 中可以用动画形式显示变量、网格、曲线等的演化过程，生成的动画文件可以用 MPEG 格式保存，并在各种媒体播放器上重放。

6.14.1 定义动画演示序列

动画演示与前面介绍的监视器显示在很多方面是类似的，主要区别是动画是动态过程，而监视器显示的内容虽然也是变化的，但是不如动画生动。

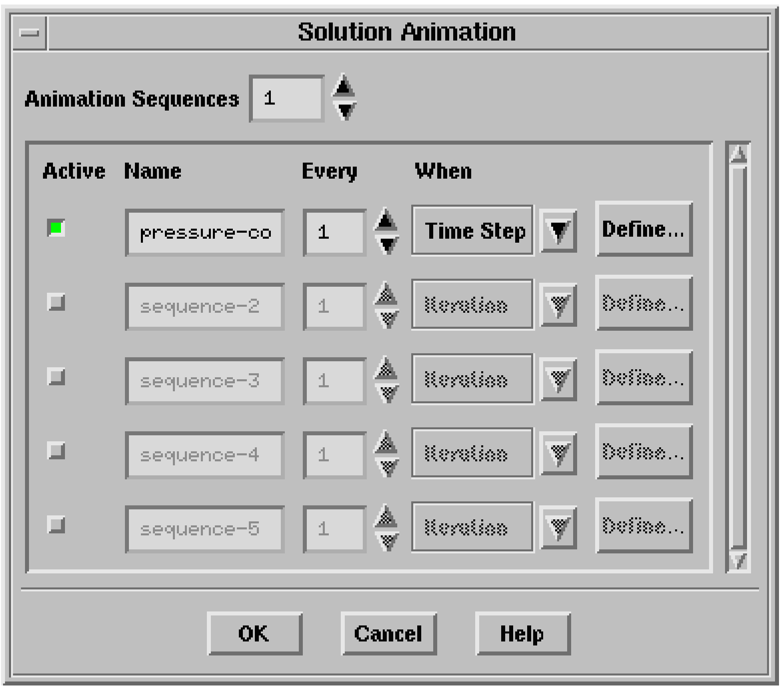


图 6-16 求解动画面板

与监视器的设置过程类似，动画的设置也是在两个面板中完成的。首先启动 Solution

Animation（求解动画）面板：

Solve->Animate->Define...

在这个面板中，可以指定创建动画的数量（Animation Sequences），然后打开 Active（激活）按钮，指定每个动画的名称及采样间隔，最后点击 Define（定义）按钮进入 Animation Sequence（动画序列）面板。

在动画序列面板中，主要设定动画的 Storage Type（保存类型）和 Display Type（显示类型）。

FLUENT 中保存动画的方式包括内存（In Memory）、图元（Metafile）和位图（PPM Image）三种方式。如果生成的动画不大，可以选择保存在内存中。如果需要将动画文件保存在硬盘上，可以选择图元或位图方式。其中图元格式的动画文件占用硬盘空间比较大，如果硬盘空间不大，最好选择位图方式。

显示类型包括网格（Grid）、等值线（Contours）、矢量（Vector）、XY 曲线（XY Plot）和监视器（Monitor）五种形式，用户可以根据所显示的变量的不同而有所变化，比如在显示适应性网格变化过程的动画中可以选择 Grid 形式；在显示压强场时可以选择 Contours 形式；在显示速度场时可以选择 Vector 形式；在显示残差变化时可以选用 XY Plot 形式等等。

在确定了显示类型后，可以点击 Properties...按钮进行详细设置，这里不再细述。

6.14.2 播放动画

Playback（动画播放）面板的启动方式为：

Solve->Animate->Playback...

如图 6-17 所示，动画播放面板与常用的媒体播放软件十分相象。在动画列表中选择一个动画文件，然后点击播放键就可以观看动画。在这个面板上可以设置播放的速度和播放的动画范围，也可以一帧一帧地进行播放。

比如计算进行了 5000 步，动画设置中设定的是每个时间步绘制一个动画帧，则最后的动画文件是由 5000 帧画面组成的。这样一个动画播放起来可能很费时，因此在第一次观看的时候可以拖动速度滑块，让动画以快速形式播放。如果在观看的过程中对某一段动画感兴趣的话，可以选择动画的开始（Start Frame）帧和结束帧（End Frame），缩小播放范围仔细观看。如果在观看过程中发现某一帧中的某个图像细节需要仔细研究的话，可以选择逐帧播放的形式，一帧一帧地仔细研究图像细节的变化。

在动画播放的过程中，可以用鼠标拖动的方式对画面进行平移、旋转、缩放等操作。如果对动画不满意，可以点击 Delete 键删除动画文件。还可以在 Play Mode（播放模式）列表中选择 Play Once（播放一次）、Auto Repeat（自动重播）、自动回放（Auto Reverse）等等。

在 Write/Record Format（写/记录格式）中，可以选择动画文件的保存格式。如果前面选择了用内存方式保存文件的话，可以在这里将内存中的动画文件转为硬盘文件。

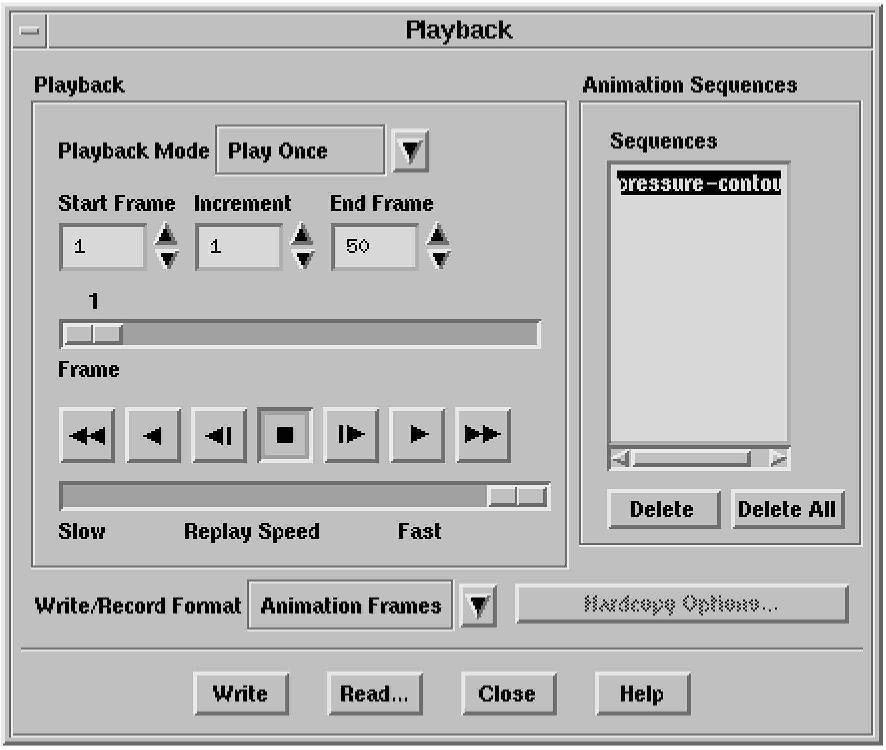


图 6-17 Playback（动画播放）面板

如果动画文件已经存在于硬盘上，则可以用播放面板直接将它读入，而不必读入相应的算例文件和数据文件。

6.15 在计算过程中执行命令

在 FLUENT 的计算过程中，每隔一段时间间隔可以执行文本命令或宏命令。这项功能在 Execute Commands（执行命令）面板中进行设置，这个面板的启动方法为：

Solve->Execute Commands...

如图 6-18 所示，在这个面板中可以指定需要执行的命令数（Defined Commands），然后激活命令栏，定义命令执行的时间间隔，再写入文本命令或宏命令的名称即可。

宏命令是一个可以自动执行的命令序列。如果需要在一定的间隔后进行一系列的命令操作，则可以先录制宏命令，然后在命令栏中填入宏命令的名称。点击面板下方的 Define Macros...（定义宏命令）按钮可以进入 Define Macro（宏命令定义）面板，如图 6-19 所示。

在宏命令定义面板中，首先设定宏命令的名称，然后点击 OK 按钮开始宏命令录制过程。在接下来的时间里，依次进行相关操作，比如让网格进行适应性调整。录制完毕后，在执行命令面板中点击 End Macro（结束宏命令）按钮结束录制过程。在后面的计算中就可以引用这个宏命令完成相关的操作。

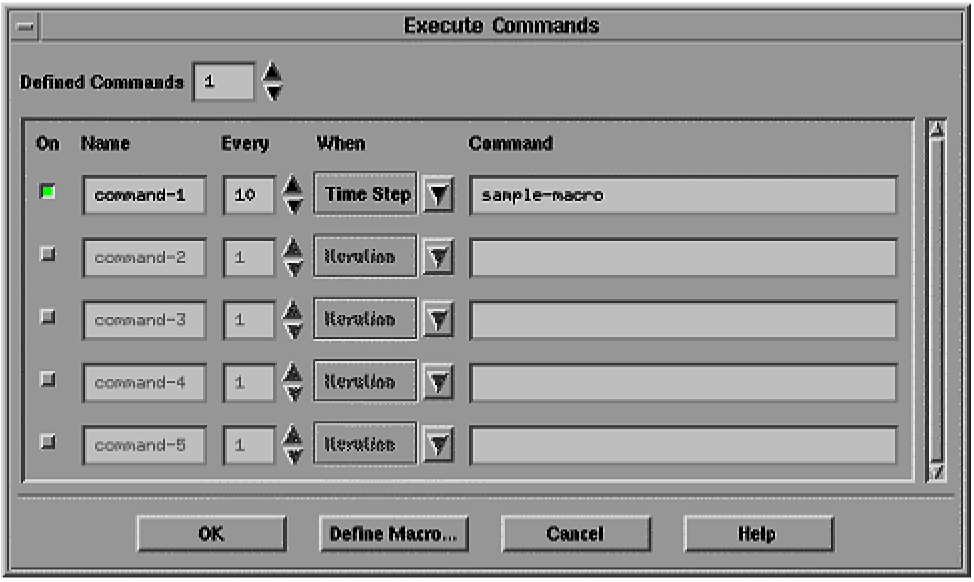


图 6-18 执行命令面板



图 6-19 宏命令定义面板

如果计算过程中需要保存算例文件和数据文件，则应该用 Autosave Case/Data（自动保存算例和数据文件）面板，其菜单操作顺序为：

File -> Write -> Autosave...

6.16 收敛性和稳定性

6.16.1 判断收敛

判断收敛性的常规方法是观察残差的变化——如果残差曲线下降则意味着计算收敛，反之则意味着发散。除此之外，还可以观察流场变量的变动情况，如果重要的流场变量在经过一段时间的计算不再变化，则意味着计算已经收敛，可以结束计算。

影响收敛性的因素很多，包括亚松弛因子、初始流场、物理模型等都会对计算是否收敛形成直接影响。正因为如此，所以很难给出一个统一的收敛判据。在大多数情况下，残差下降 3 个数量级就可以认为计算已经收敛了，但是如果初始流场非常靠近最终解，同时残差计算以初始场为基准进行比例计算，则残差下降的幅度可能不会达到 3 个数量级。与此相反，如果初始流场设置的非常粗糙，则在达到收敛时残差的下降幅度可能又远远超过 3 个数量级。因此，在用残差下降幅度判断收敛之外，还可以用残差在计算过程中的行为进行收敛性判断，即残差持续下降时间或迭代步数达到一定的值，就可以认为计算已经收敛了。另外，因为在定常计算中，"收敛"的真实含义就是流场变量的变化幅度已经足够小，以致我们可以认为流场不再变化，所以也可以直接用流场变量的变化幅度作为计算是否收敛的判据。

6.16.2 加速收敛

在计算发散时最容易想到的解决办法是减小亚松弛因子的值，在很多情况下，这个办法是有效的，因为计算发散的最常见原因是由方程的非线性引起的，而亚松弛因子是用线性稳定性分析得到的，因此减小亚松弛因子可以让计算回到稳定域中。

计算发散还可能由初始流场的设置引起，即如果初始流场给的过分粗糙，也有可能引起计算的发散。在这种情况下，解决问题的办法当然应该从改善初始流场入手。如果计算中涉及的物理模型非常复杂，可以考虑先计算简化流场，再以简化流场为初始流场继续进行计算。比如，在多相流计算中，可以先计算连续相流场，然后再加入弥散相进行计算。在带有化学反应的流场计算中，可以先计算混合物流场，或冻结流流场，再计算带化学反应模型的流场，等等。

6.16.3 修改多步时间推进参数

控制多步时间推进格式最常用的参数是库朗（Courant）数，除此之外，还可以通过残差光顺和改变多步格式的方法增加计算稳定性或增加时间步长。

用残差光顺的方法增加库朗数

在 Solution Controls（求解过程控制）面板中，残差光顺的迭代值在缺省设置中被设定为 0，即在缺省设置中没有使用残差光顺技术。如果将 Iterations（迭代计数器）增加为 1 或更大的数，则可以进一步设置 Smoothing Factor（光顺因子）。将光顺因子设定为 0.5 可以将库朗数增加为原数值的两倍。

改变多步格式

首先启动 Multi-Stage Parameters（多步格式参数）面板：

Solve->Controls->Multi-Stage...

在这个面板中，可以增加多步格式的步数，重新设置每一步的系数，并决定是否每步更新耗散项和粘性应力的值。

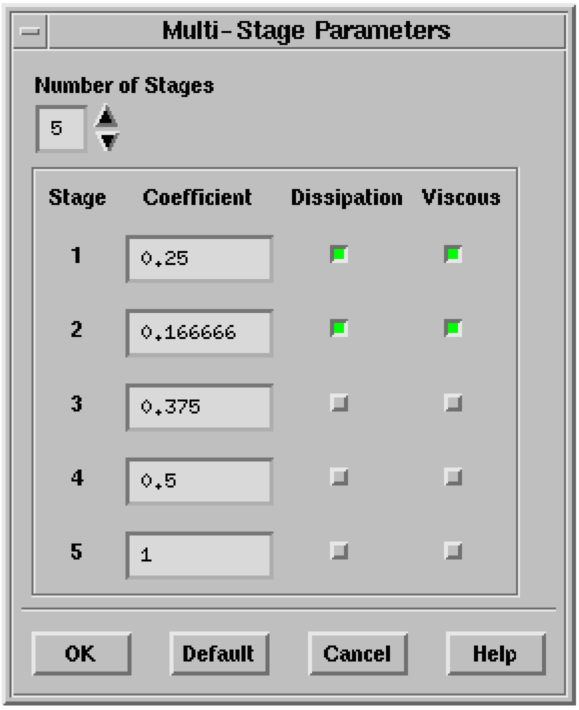


图 6-20 多步格式参数面板

在缺省设置中，FLUENT 使用 5 步格式，每步的系数分别为 0.25、0.166666、0.375、0.5 和 1.0。在对多步格式非常熟悉的情况下可以增加多步格式的步数，同时修改每步的系数。修改系数的一般要求是：

（1）系数为介于 0 和 1 之间的实数。

（2）最后一步的系数必须为 1。

在多步格式参数面板上还可以决定是否在每一步都计算耗散项和粘性项。选中耗散项

意味着在相应的计算步中需要计算人工耗散，如果不选，则在相应的计算步中人工耗散项保持与前一步相同的值，即人工耗散项被“冻结”。粘性项的情况与此类似，在被选中时计算粘性应力，不选时，粘性应力被冻结。需要注意的是，在第一步计算中应该计算粘性应力，随后的计算中如果继续计算粘性应力可以增加计算的鲁棒性，但同时需要消耗更多的 CPU 时间。

在希望恢复缺省设置时，可以点击 Default（缺省）按钮，此时所有参数值恢复缺省值，同时 Default（缺省）按钮变为 Reset（重置）按钮。如果希望将缺省值恢复为设定的值，可以点击 Reset 按钮，则所有参数值可以重新恢复人为修改过的参数值。

第七章 FLUENT 的物理模型

FLUENT6.1 中采用的物理模型包括基本流动模型、湍流模型、动网格模型、化学反应模型、燃烧模型、PDF 模型、弥散相模型、多相流模型、热交换模型、气动噪声模型和固化与熔化模型等等，可以模拟包括可压、不可压、定常、非定常等流动范围的问题，涉及的工业应用领域包括加工设备中的层流非牛顿流体问题、旋转机械和风扇问题、空气动力学内外流问题、多相流问题、多孔介质问题、气动噪声问题等等。限于本书篇幅，本章仅介绍与软件使用相关的内容，更深入的理论知识，请读者参阅有关专业方向的参考资料。

7.1 基本流动模型

FLUENT 在所有问题的求解过程中都会求解质量和动量的守恒方程。在涉及到可压缩性和热交换的问题时则求解能量方程。如果计算问题中包含组分输运过程，比如在化学反应流，或混合物流动的计算中，则增加对组分浓度方程的求解。在包含湍流模型的问题中，还会增加对湍流动能等湍流变量的守恒方程的求解。

7.1.1 周期性流动

周期性流动即流动中包含周期性特征的流动。周期性流动特征则是在一定的时间、空间间隔上流场变量或其导数重复出现的现象。当然从严格的数学意义上说，周期性流动中流场变量或其导数值可能只是在一定的时间、空间间隔上出现近似解，而不一定是绝对相等的解。如果严格按照数学定义来定义周期性，显然会增加软件设计的难度，因此 FLUENT 中所说的周期性是绝对相等的变量或其增量在一定时间、空间间隔上重复出现的现象。

如边界条件一章中所述，周期性边界条件包括旋转周期性条件和平移边界条件，即沿圆周方向呈周期性流动特点，或沿直线方向呈周期性流动特点。周期性流动的另一个分类方法是看流动过程中是否存在压力降，即有压力降的周期性流动和无压力降的周期性流动。本章讨论的流向周期性流动（streamwise periodic flows），指的是沿流动方向存在周期性特征，同时不存在压力降的情况。在周期性流动中，相隔固定空间距离的速度分量和压强的值是相等的。

在周期性计算中，流动必须满足的条件包括：

（1）流动为不可压流。

（2）流场在流动方向上的几何特征具有周期性特征。

（3）如果采用耦合求解器，只能给定压强增量。如果采用分离求解器，则可以同时给

定压强增量和质量流量。

（4）在流场入口和出口上没有质量的净增量。

（5）弥散相模型、多相流模型和反应流模型不能出现在计算中。

在采用分离求解器时，周期性流动计算的设置面板为周期性条件（Periodicity Conditions）面板：

Define->Periodic Conditions...

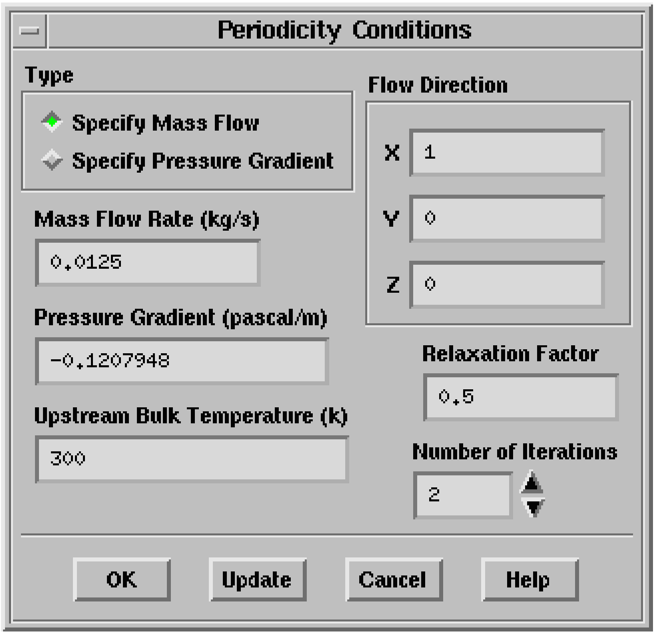


图 7-1 周期性条件面板

设定过程中首先选择采用质量流（Specify Mass Flow）或压强梯度（Specify Pressure Gradient）定义方式。然后相应地在下面栏目中输入质量流或压强梯度的值，并定义流场温度和方向。假设采用质量流方式，则压强梯度要通过计算获得，此时定义松弛因子（Relaxation Factor）和迭代次数（Number of Iteratons），就可以通过系统计算出压强梯度的值。

在采用耦合求解器时，应该在周期性（Periodic）面板中进行设置：

Define->Boundary Conditions...

如图 7-2 所示，在周期性面板中在选定的区域上设定周期性计算的类型，即平移型还是旋转型，同时给出周期性压强增量的值，点击 OK 按钮即可。

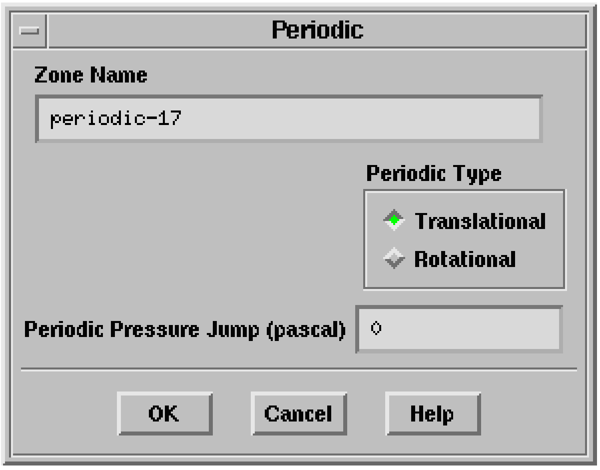


图 7-2 周期性面板

7.1.2 旋转流动

旋转流动是工程应用方面常见的问题类型之一，其流场特征就是流场是由几个大的集中涡组成。在旋转流动中，$k-\varepsilon$ 湍流标准模型失效，因此需要采用 RNG $k-\varepsilon$ 模型、realizable $k-\varepsilon$ 模型做湍流计算。在旋转特别强烈时，建议采用 RSM 模型计算湍流。旋转流动对网格精度要求较高，因此精细的网格对于提高旋转流动的计算精度是必不可少的。在计算轴对称问题时需要注意的是，FLUENT 中规定只能用 X 轴作对称轴，同时网格必须位于 X 轴的上方，即 Y 坐标大于 0 的区域。

轴对称计算需要在求解器（Solver）面板和边界条件（Boundary Conditions）面板中进行设置，即在边界条件面板中的 Space 选项中打开 Axisymmetric Swirl，同时在入口或壁面边界条件中定义旋转边界的转动速度。

求解轴对称旋转流动与求解其他类型的复杂流动问题一样，都可以采取从简单到复杂的计算策略，即为了保证计算的稳定性，可以先计算旋转不太强烈的流场，再逐渐增加旋转强度的方式，逐渐增加流场复杂性，以便保证最后得到高精度的结果。

在后处理过程中可以报告旋转速度（Swirl Velocity）和旋转壁面的剪切应力（Swirl-Wall Shear Stress）。

7.1.3 可压缩流

可压缩流计算通常用于气流流动计算，液体流动在绝大多数情况下被认为是不可压的。衡量流体可压缩性的指标是马赫数。在马赫数小于 0.3 时，气体通常被认为是不可压流体。在马赫数大于 0.3 时，气体的压缩性影响逐渐增强，此时计算中必须考虑压缩性影响。

可压缩流计算中各项参数的设置过程与其他流动计算大同小异，主要区别是在计算中

引入了粘性加热效应，即需要在定义粘性影响和湍流模型的 Viscous 面板中，打开 Viscous Heating 选项，同时需要在物质属性设置中打开理想气体选项，即在 Material 面板的 Density 旁边打开 Ideal-gas 选项。

在计算域中的压强随时间缓慢增长时，可以在操作条件中打开浮动操作压强（floating operating pressure）选项以减小计算中的舍入误差。需要注意的是，这个选项在使用压强入口、压强出口、排气风扇、进气风扇、吸入式风扇、通风出口或压强远场条件等边界条件时不能使用。

可压缩流计算中需要设定的边界条件如下：

（1）在流场入口：

- 压强入口条件：给定入口总温、总压，在超音速入口时，还需设定静压。
- 质量流入口：给定入口质量流和总温。

（2）在流场出口：

- 压强出口条件：给定出口静压。

在计算结束后，可以报告的项目包括总温、总压和马赫数等参数。

7.1.4 无粘流

无粘流假设适用于粘性对流场影响可以忽略的计算中，比如飞行器的气动计算中，如果仅需要计算升力的话，则可以考虑适用无粘流假设计算。无粘流计算的另一个用途是为复杂流动计算提供初始流场。

无粘流计算的设置是在 Viscous 面板中进行的，在面板中选择 Inviscid 选项，并在边界条件和物质属性中进行相应的设置就可以进行计算了。

在后处理过程中，可以报告物体的受力：

Report->Forces...

7.2 湍流模型

因为湍流现象是高度复杂的，所以至今还没有一种方法能够全面、准确地对所有流动问题中的湍流现象进行模拟。在涉及湍流的计算中，都要对湍流模型的模拟能力以及计算所需系统资源进行综合考虑后，再选择合适的湍流模型进行模拟。FLUENT 中采用的湍流模拟方法包括 Spalart-Allmaras 模型、standard（标准）$k-\varepsilon$ 模型、RNG（重整化群）$k-\varepsilon$ 模型、Realizable（现实）$k-\varepsilon$ 模型、v^2-f 模型、RSM（Reynolds Stress Model，雷诺应力模型）模型和 LES（Large Eddy Simulation，大涡模拟）方法。本小节将介绍各种湍流计算方法的特点与使用限制，为在计算中正确选择湍流计算方法提供简单的背景知识。更多的与湍流相关的内容还要参考相关领域的文献资料。

7.2.1 雷诺平均与大涡模拟的对比

因为直接求解 NS 方程非常困难，所以通常用两种办法对湍流进行模拟，即对 NS 方程进行雷诺平均和滤波处理。这两种方法都会增加新的未知量，因此需要相应增加控制方程的数量，以便保证未知数的数量与方程数量相同，达到封闭方程组的目的。

雷诺平均 NS 方程是流场平均变量的控制方程，其相关的模拟理论被称为湍流模式理论。湍流模式理论假定湍流中的流场变量由一个时均量和一个脉动量组成，以此观点处理 NS 方程可以得出雷诺平均 NS 方程（简称 RNS 方程）。在引入 Boussinesq 假设，即认为湍流雷诺应力与应变成正比之后，湍流计算就归结为对雷诺应力与应变之间的比例系数（即湍流粘性系数）的计算。根据计算中使用的变量数目和方程数目的不同，湍流模式理论中所包含的湍流模型又被分为二方程模型、一方程模型和零方程模型（代数模型）等大类。FLUENT 中使用的三种 $k-\varepsilon$ 模型、Spalart-Allmaras 模型、$k-\omega$ 模型及雷诺应力模型（RSM）等都属于湍流模式理论。

大涡模拟（LES）方法是通过滤波处理计算湍流的，其主要思想是大涡结构（又称拟序结构）受流场影响较大，小涡则可以认为是各向同性的，因而可以将大涡计算与小涡计算分开处理，并用统一的模型计算小涡。在这个思想下，大涡模拟通过滤波处理，首先将小于某个尺度的旋涡从流场中过滤掉，只计算大涡，然后通过求解附加方程得到小涡的解。过滤尺度一般就取为网格尺度。显然这种方法比直接求解 NS 方程的 DNS 方程效率更高，消耗系统资源更少，但却比湍流模式方法更精确。尤其应该注意的是，湍流模式理论无法准确模拟大涡结构，因此在需要模拟大涡结构时，只能采用 LES 方法[1]。

尽管大涡模拟理论比湍流模式理论更精确，但是因为大涡模拟需要使用高精度的网格，对计算机资源的要求比较高，所以还不能在工程计算中被广泛使用。在绝大多数情况下，湍流计算还要采用湍流模式理论，大涡模拟则可以在计算资源足够丰富的时候尝试使用。

7.2.2 Spalart-Allmaras 模型

Spalart-Allmaras 模型是一方程模型里面最成功的一个模型，最早被用于有壁面限制情况的流动计算中，特别在存在逆压梯度的流动区域内，对边界层的计算效果较好，因此经常被用于流动分离区附近的计算，后来在涡轮机械的计算中也得到广泛应用。

最早的 Spalart-Allmaras 模型是用于低雷诺数流计算的，特别是在需要准确计算边界层粘性影响的问题中效果较好。FLUENT 对 Spalart-Allmaras 进行了改进，主要改进是可以在网格精度不高时使用壁面函数。在湍流对流场影响不大，同时网格较粗糙时，可以选用这个模型。

Spalart-Allmaras 模型是一种新出现的湍流模型，在工程应用问题中还没有出现多少成功的算例。如同其他一方程模型一样，Spalart-Allmaras 模型的稳定性也比较差，在计算中

[1] FLUENT 中尚未提供 DNS 计算选项。

采用 Spalart-Allmaras 模型时需要注意这个特点。

7.2.3 标准$k-\varepsilon$模型

标准$k-\varepsilon$模型由 Launder 和 Spalding 提出，模型本身具有的稳定性、经济性和比较高的计算精度使之成为湍流模型中应用范围最广、也最为人熟知的一个模型。标准$k-\varepsilon$模型通过求解湍流动能（k）方程和湍流耗散率（ε）方程，得到k和ε的解，然后再用k和ε的值计算湍流粘度，最终通过 Boussinesq 假设得到雷诺应力的解。

虽然得到了最广泛的使用，但因为标准$k-\varepsilon$模型假定湍流为各向同性的均匀湍流，所以在旋流（swirl flow）等非均匀湍流问题的计算中存在较大误差，因此后来又发展出很多$k-\varepsilon$模型的改进模型，其中包括 RNG（重整化群）$k-\varepsilon$模型和 Realizable（现实）$k-\varepsilon$模型等衍生模型。

7.2.4 RNG $k-\varepsilon$模型

RNG $k-\varepsilon$模型在形式上类似于标准$k-\varepsilon$模型，但是在计算功能上强于标准$k-\varepsilon$模型，其改进措施主要有：

（1）在ε方程中增加了一个附加项，使得在计算速度梯度较大的流场时精度更高。

（2）模型中考虑了旋转效应，因此对强旋转流动计算精度也得到提高。

（3）模型中包含了计算湍流 Prandtl 数的解析公式，而不象标准$k-\varepsilon$模型仅用用户定义的常数。

（4）标准$k-\varepsilon$模型是一个高雷诺数模型，而重整化群$k-\varepsilon$模型在对近壁区进行适当处理后可以计算低雷诺数效应。

7.2.5 Realizable $k-\varepsilon$模型

Realizable $k-\varepsilon$模型与标准$k-\varepsilon$模型的主要区别是：

（1）Realizable $k-\varepsilon$模型中采用了新的湍流粘度公式。

（2）ε方程是从涡量扰动量均方根的精确输运方程推导出来的。

现实$k-\varepsilon$模型满足对雷诺应力的约束条件，因此可以在雷诺应力上保持与真实湍流的一致。这一点是标准$k-\varepsilon$模型和 RNG $k-\varepsilon$模型都无法做到的。这个特点在计算中的好处是，可以更精确地模拟平面和圆形射流的扩散速度，同时在旋转流计算、带方向压强梯度的边界层计算和分离流计算等问题中，计算结果更符合真实情况。

Realizable $k-\varepsilon$模型是新出现的$k-\varepsilon$模型，虽然还无法证明其性能已经超过 RNG $k-\varepsilon$模型，但是在分离流计算和带二次流的复杂流动计算中的研究标明，Realizable $k-\varepsilon$模型是所有$k-\varepsilon$模型中表现最出色的湍流模型。

Realizable $k-\varepsilon$ 模型在同时存在旋转和静止区的流场计算中，比如多重参考系、旋转滑移网格等计算中，会产生非物理湍流粘性，因此在类似计算中应该慎重选用这种模型。

7.2.6 $k-\omega$ 模型

$k-\omega$ 模型也是二方程模型。标准 $k-\omega$ 模型中包含了低雷诺数影响、可压缩性影响和剪切流扩散，因此适用于尾迹流动计算、混合层计算、射流计算，以及受到壁面限制的流动计算和自由剪切流计算。

剪切应力输运 $k-\omega$ 模型，简称 SST $k-\omega$ 模型，综合了 $k-\omega$ 模型在近壁区计算的优点和 $k-\varepsilon$ 模型在远场计算的优点，将 $k-\omega$ 模型和标准 $k-\varepsilon$ 都乘以一个混合函数后再相加就得到这个模型。在近壁区，混合函数的值等于 1，因此在近壁区等价于 $k-\omega$ 模型。在远离壁面的区域混合函数的值则等于 0，因此自动转换为标准 $k-\varepsilon$ 模型。

与标准 $k-\omega$ 模型相比，SST $k-\omega$ 模型中增加了横向耗散导数项，同时在湍流粘度定义中考虑了湍流剪切应力的输运过程，模型中使用的湍流常数也有所不同。这些特点使得 SST $k-\omega$ 模型的适用范围更广，比如可以用于带逆压梯度的流动计算、翼型计算、跨音速激波计算等等。

7.2.7 v^2-f 模型

v^2-f 模型与 $k-\varepsilon$ 模型比较类似，但是 v^2-f 模型中考虑到了壁面附近湍流的各向异性问题和非局部的压强与应变的关系。v^2-f 模型属于低雷诺数湍流模型，其适用范围从自由流区一直延伸到壁面，并且无需适用壁面函数。v^2-f 模型主要用于边界层计算和分离流计算。

v^2-f 模型的突出特点是用速度尺度 $\overline{v^2}$ 代替湍流动能 k 计算湍流粘度。速度尺度 $\overline{v^2}$ 代表速度的脉动量，因此可以更准确地模拟湍流变量在壁面附近的耗散过程。这点是用湍流动能 k 进行计算时无法做到的。

7.2.8 雷诺应力模型（RSM）

雷诺应力模型中没有采用涡粘度的各向同性假设，因此从理论上说比湍流模式理论要精确得多。雷诺应力模型不采用 Boussinesq 假设，而是直接求解雷诺平均 NS 方程中的雷诺应力项，同时求解耗散率方程，因此在二维问题中需要求解 5 个附加方程，在三维问题

中则需要求解 7 个附加方程。

从理论上说，雷诺应力模型应该比一方程模型和二方程模型的计算精度更高，但实际上雷诺应力模型的精度受限于模型的封闭形式，因此雷诺应力模型在实际应用中并没有在所有的流动问题中都体现出其优势。只有在雷诺应力明显具有各向异性的特点时才必须使用雷诺应力模型，比如龙卷风、燃烧室内流动等带强烈旋转的流动问题。

7.2.9 湍流模型的计算速度

计算速度的快慢与计算量成反比，即计算量大则计算速度慢，需要的时间也长。湍流模型计算中的工作量主要取决于方程的数量和方程中函数项的多少。如果不考虑大涡模拟方法，湍流模型计算从总体上说，一方程模型（Spalart-Allmaras 模型）计算最快，二方程（$k-\varepsilon$ 模型、$k-\omega$ 模型、v^2-f 模型）模型次之，雷诺应力模型最慢。

7.2.10 壁面函数和近壁模型

在受壁面限制的流动中，因为壁面附近流场变量的梯度较大，所以壁面对湍流计算的影响很大。湍流模型中假定湍流是各向同性的，因此在壁面附近需要进行特殊处理。处理的一种办法是用半经验公式将自由流中的湍流与壁面附近的流动连接起来，这种方法被称为壁面函数法。另一种方法是通过在壁面附近加密网格，同时调整湍流模型以包含壁面影响的方法，被称为近壁模型法。

壁面函数法中又有标准壁面函数法和非平衡壁面函数法。一般地说，标准壁面函数可以适用于大多数流动问题，因此也是 FLUENT 中缺省设置的方法。非平衡壁面函数法则适用于流场函数在壁面附近存在很大梯度的流动问题。

壁面函数法适用于高雷诺数流动，近壁模型法适用于低雷诺数流动。

7.2.11 湍流计算方法的设置

湍流计算方法的选择在 Viscous（粘性）面板中进行。在确定了所需要的湍流模型之后，逐项进行选择即可，这里不再重复其操作过程。

7.3 活动变形区域中的流动计算

在计算单一运动物体内部或外部的流场时，可以通过将坐标系定义在物体上的办法，将问题转化为物体静止而来流变化的流动问题。比如在计算搅拌器内的流动时，就可以用

这种办法将参考坐标系连接在搅拌轮上，而将搅拌桶定义成动边界。从这个坐标系看去，搅拌轮是静止的，而搅拌桶是转动的，因而通过坐标转化可以用固定网格完成流动计算。

然而在多个物体之间存在相对运动时，简单地转换参考系显然是不能解决问题的。比如在计算涡轮发动机中的静子与转子的干扰问题时，无论怎样设置参考系，都会遇到固体边界随时间变化的问题。FLUENT 中用于这类问题计算的模型包括多重参考系（Multiple Reference Frame, MRF）模型、混合面（Mixing Plane）模型、滑移网格（Sliding Meshes）技术。如果固体边界不是发生平动或转动，而是随时间变化发生形变，则可以用动态网格（Dynamic Meshes）技术进行计算。

7.3.1 旋转坐标系中的流动计算

采用旋转坐标系进行流动计算时，需要将坐标系与外形复杂的部件连接在一起，而被设置为做旋转运动的部件，在几何上必须是以旋转中心为圆心形成的一个圆或圆柱。这样可以将问题转化为定常流动问题，从而大大简化计算。比如在前面提到的搅拌器中，搅拌桶是一个以搅拌轴为对称轴的圆柱体，搅拌轮则是一个带叶片的复杂几何体。如果将坐标系与搅拌轮相连，则只要在 FLUENT 中将作为边界的搅拌桶壁面设为运动边界，同时将其转动角速度取为搅拌轮角速度的负值，则整个问题就可以被简化为定常问题。反之，如将坐标系与搅拌桶相连接，则会使问题复杂化。

网格设置的注意事项

在采用旋转坐标系进行计算时，在网格使用上 FLUENT 规定：

（1）在二维问题中，旋转轴必须平行于 Z 轴，即垂直于屏幕所在平面。

（2）在轴对称问题中，计算域取为过旋转轴的一个二维平面区域，此时必须将旋转轴设为 X 轴。

（3）在三维问题中，旋转轴可以取为任意直线，但是为方便起见，最好还是将旋转轴设为某个坐标轴。

旋转坐标系的设定过程

采用旋转坐标系计算时，相关设置过程为：

（1）在 Solver（求解器）面板中的 Velocity Formulation（速度公式）项目下选择 Relative（相对速度）或 Absolute（绝对速度）。因为在用耦合求解器时只能使用绝对速度，所以不用设置此项目。

（2）在 Boundary Conditions（边界条件）面板中，从边界列表中选择旋转边界，然后点击 set（设置）按钮打开 Fluid（流体）面板或 Solid（固体）面板进行边界条件设置，包括 Rotation-Axis Origin（旋转轴原点）和 Rotation-Axis Direction（旋转轴方向）的设置，同时 Motion Type（活动类型）中选择 Moving Reference Frame（活动参考系），然后在 Rotational Velocity（旋转速度）下设置 Speed（速度）。

（3）定义壁面速度。

（4）在速度入口处定义速度，在压强入口定义流动方向和总压。

总压和流动方向的定义都取决于速度的定义方式，即如果使用绝对速度定义，总压和流动方向均在绝对坐标系中定义；在使用相对速度定义时，总压和流动方向则相对于旋转坐标系定义。

在压强出口，静压定义与坐标系无关。在回流区，无论采用绝对速度还是相对速度定义，给定的静压值都被用作回流的总压，而流动方向都被假定垂直于压强出口边界。

求解策略

- 在旋转强烈的区域，建议使用相对速度定义。在其余区域，建议使用绝对速度定义。
- 建议使用 PRESTO!格式。
- 在网格划分时，应该在压强、速度梯度较大的区域加密。
- 降低亚松弛因子，比如可以将亚松弛因子调整为 0.3-0.5，甚至更低。
- 开始计算的时候可以从较低的旋转速度算起，然后逐渐增加旋转速度。

7.3.2 多重参考系（MRF）模型

如前所述，在计算域中同时存在动网格区和静网格区时，可以采用的模型包括 MRF 模型、混合面模型和滑移网格模型。其中 MRF 模型和混合面模型用于定常流计算，而 MRF 则是最简单的，因而也是最经济的模型。

在使用 MRF 模型进行计算时，整个计算域被分成多个小的子域。每个子域可以有自己的运动方式，或静止，或旋转，或平移。流场控制方程在每个子域内进行求解，在子域的交界面上则通过将速度换算成绝对速度的形式进行流场信息交换。

如果交界面上网格是正则的，即交界面两侧区域共用相同的网格节点，则网格划分方法与常规方法别无二致。如果网格是非正则的，其处理方法请参阅本书中与非正则网格处理相关的内容。

MRF 模型的设置

与旋转坐标系的设置类似，MRF 模型的设置方法为：

（1）在 Solver（求解器）面板中选择速度的定义形式，即决定采用相对速度定义，还是绝对速度定义。当然在使用耦合求解器时，不用进行此项设置。

（2）在 Boundary Conditions（边界条件）面板中选择相应的边界名称，点击 set（设置）按钮进入 Fluid（流体）面板或 Solid（固体）面板，并设置相应的旋转轴原点和旋转轴方向。在 Motion Type（运动类型）列表中选择 Moving Reference Frame（移动参考系），并设置 Rotational Velocity（旋转速度）下的 Speed（速度），或 Translational Velocity（平动速度）的 X、Y、Z 分量。

（3）在壁面上设置速度。

（4）在速度入口定义速度，在压强入口定义总压和流动方向。

注意事项

（1）最好不要与 Realizable $k-\varepsilon$ 模型配合使用。

（2）活动网格区域在与相邻区域的交界面上的法向速度应该为零。

（3）最好将 MRF 模型的使用限制在定常流计算中，而将非定常流计算交给后面介绍的滑移网格模型进行计算。

（4）在采用相对速度定义时，无质量粒子的迹线显示是有意义的，而有质量粒子，包括弥散相粒子的迹线显示是无意义的。因此在需要跟踪粒子轨迹的时候，最好采用绝对速度定义。

（5）在采用相对速度定义时，MRF 对轴对称旋流的计算不够精确。

（6）MRF 模型中的平移速度和旋转速度都是常数，即边界上速度应该是“均匀”的。

7.3.3 混合面模型

在 MRF 模型的计算中，边界上的速度必须是相同的，否则得出的结果可能是没有意义的。在边界上各点的速度并不相同时，当然可以采用滑移网格法进行计算，但是考虑到滑移网格法消耗的系统资源较大，所以在一些情况下，混合面模型可以作为一个很好的补充，用于进行这类计算。

基本思想

与 MRF 模型一样，混合面模型也将计算域划分为数个彼此独立的子域。与 MRF 模型不同的是，混合面模型在子域的交界面上构造了一个混合面，即用面积平均的方法将交界面两侧的流场变量联系起来，而每个子域内的流场则可以作为定常流场进行求解。

例如，在多级压气机的计算中，压气机由多级静子和转子组成，每级的叶片数量都不一样。在这种情况下，可以将每级静子或转子与相邻流场区域划分出来作为计算的子域，而在交界面上用混合面相连接。这样每个划分出来的子域都可以采用不同的坐标系和边界条件，并转化为定常问题进行计算。子域间的信息交流则通过混合面完成。

混合面模型的计算过程为，首先计算各子域内的定常流场，其次计算混合面上的变量分布，然后以混合面上的变量分布为边界条件开始下一个迭代步的计算，直到计算收敛。

混合面模型的设置

（1）在 Solver（求解器）面板中设定速度的定义形式，即确定采用相对速度还是绝对速度定义。

（2）设定旋转轴原点和方向，同时设定旋转速度。

（3）设定边界上的速度。

（4）设定速度入口和压强入口条件。

（5）在 Mixing Planes（混合面）面板中定义混合面：

Define->Mixing Planes...

- 指定混合面的上、下游计算子域，即在 Upstream Zone（上游区域）和 Downstream Zone（下游区域）中指定上、下游子域的名称。
- 在三维问题中，还需要指定混合面的几何形状，即指定混合面为径向（Radial）平面，还是轴向（Axial）平面。径向平面的含义是，混合面包含旋转轴直线，流量在圆周方向进行平均。轴向平面的含义是，混合面与旋转轴相互垂直，流量沿轴向进行平均。

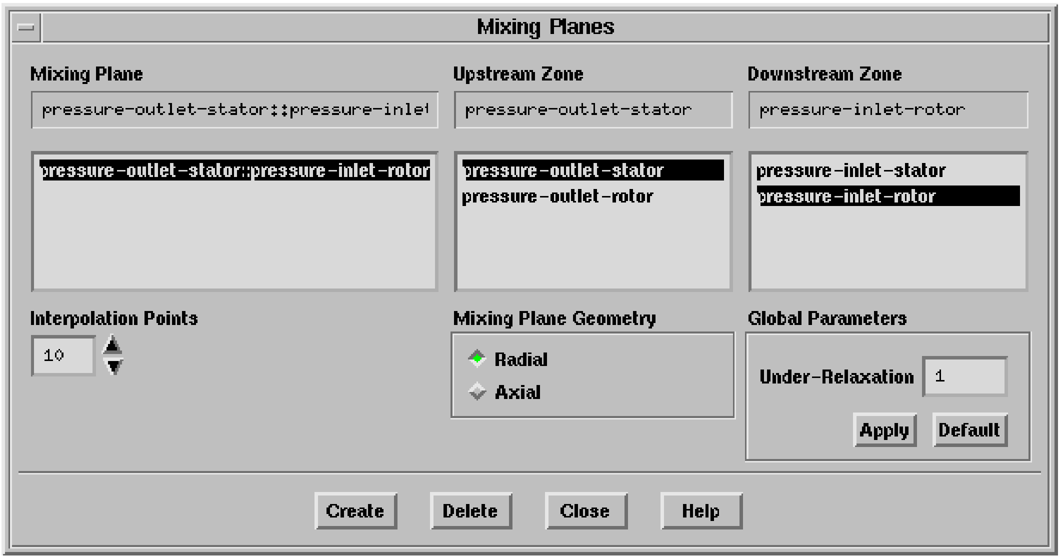


图 7-3 混合面面板

- 设定 Interpolation Points（插值点）的值，这个值是用于构造变量分布曲线的插值点数目。
- 设定 Global Parameters（全局参数），即设定混合面计算的亚松弛因子。
- 点击 Create（创建）按钮，完成混合面的创建过程。

混合面模型的选项

（1）在不可压流计算中，为了控制压强水平，可以用文本命令 fix-pressure-level（固定压强水平）：

define->mixing-planes->set->fix-pressure-level

启用这个选项后，在每次迭代结束后，FLUENT 都会从表压中减去最靠近参考压强位置的网格单元上的压强值。参考压强位置在 Operating Conditions（操作条件）面板中的

Reference Pressure Location（参考压强位置）选项中设置。

（2）用文本命令 conserve-swirl（旋流保持）可以保持涡量在混合面前后的守恒：

define->mixing-planes->set->conserve-swirl

用文本命令 enable?可以启动这项设置。启动后可以让求解器在计算过程中报告与涡量变换的相关信息。如果同时使用命令 verbosity?，FLUENT 会在每次迭代时报告指定区域的涡量，包括涡量在上、下游区域上的积分值，以及进行修正前，上、下游涡量积分的比例。如果想得到各压强入口、速度入口、质量入口的涡量积分报告，可以使用命令 report-swirl-integration（旋流积分报告）。

（3）混合面上可以加入的另一个选项是在混合面前后可以保证总焓守恒。因为对于压气机之类的涡轮机械而言，总体效率是直接与每级叶片前后总焓的改变相关的。保持总焓守恒的方法就是调整下游温度分布函数，以使其积分值与上游相等。

总焓守恒可以用文本命令 conserve-total-enthalpy 进行设定：

define->mixing-planes->set->conserve-total-enthalpy

同样，开启总焓守恒选项也必须使用 enable?命令。开启后，也可以让求解器在计算过程中报告总焓守恒信息。如果开启 verbosity?，FLUENT 将在每个迭代步中报告上、下游的热流通量及上、下游热流通量的比值。

计算策略

混合面方程在出现回流时会出现较大误差，为避免在混合面上产生回流，可以在计算开始阶段采用固定参数进行计算，等流场稳定下来后再恢复正常条件。另外，采用较低的亚松弛因子也对计算的正常进行有帮助。

7.3.4 滑移网格技术

滑移网格技术可以处理非定常问题，这是它与 MRF 模型和混合面模型的最大区别。滑移网格技术处理的通常是带有周期性的问题，比如涡轮机械中转子和静子的相互干扰问题，但是滑移网格也可以计算非周期性问题，比如两列火车交错行驶过程中周围流场的变化。在不需要考虑转子和静子相互干扰的细节时，用 MRF 模型和混合面模型进行计算就可以获得相互干扰的平均效果，但是在需要考虑干扰过程的细节时，则必须使用滑移网格技术。滑移网格技术在计算中需要使用的系统资源比较大，因此在使用滑移网格技术时需要使用配置较高的计算机。

基本思想

在滑移网格计算中，计算域至少包含 2 个以上存在相对运动的子域。每个运动子域至少有一个与相邻子域连接的交界面。原则上交界面形状是任意的，但在实际计算中，交界面的实际形状都设计成在滑移后相邻子域不能相互重叠的形状，或者说交界面上的运动速度必须与交界面相垂直。比如在旋转机械问题中，交界面都设计成轴对称形式，包括圆锥

面、圆柱面等形状。在列车交错问题中，交界面则设计成平面等等。

滑移网格技术中设定的交界面在计算过程中总是有一部分与相邻子域相连，而其余区域则不与相邻子域相连。与相邻子域相连的区域被称为内部区域。与相邻子域不相连的区域，在平动问题中被称为壁面区域，在周期性流动问题中则被称为周期区域。在每次迭代结束后，FLUENT 都会重新计算内部区域的范围，将交界面的其余部分划定为壁面区域或周期性区域，并在壁面区域和周期性区域上则设定相应的边界条件。在新的迭代步中，只计算内部区域上的通量。滑移网格计算中采用非正则网格技术，即交界面两侧子域在交界面上不共用网格节点，因此内部区域不是用交界面两侧的网格面直接构成的，而是通过子域间的相对移动量重新计算得出内部区域的边界位置。

滑移网格技术的设置和求解

滑移网格技术的设置过程如下：

（1）在求解器面板中将计算类型设置为非定常。

（2）在边界条件面板中，将交界面的网格类型设置为 interface。同时在 Fluid 面板和 Solid 面板中，将移动区域的运动类型（Motion Type）设置为移动网格（Moving Mesh），并设定其移动速度。

（3）在 Grid Interfaces 面板上定义网格交界面：

Define->Grid Interfaces...

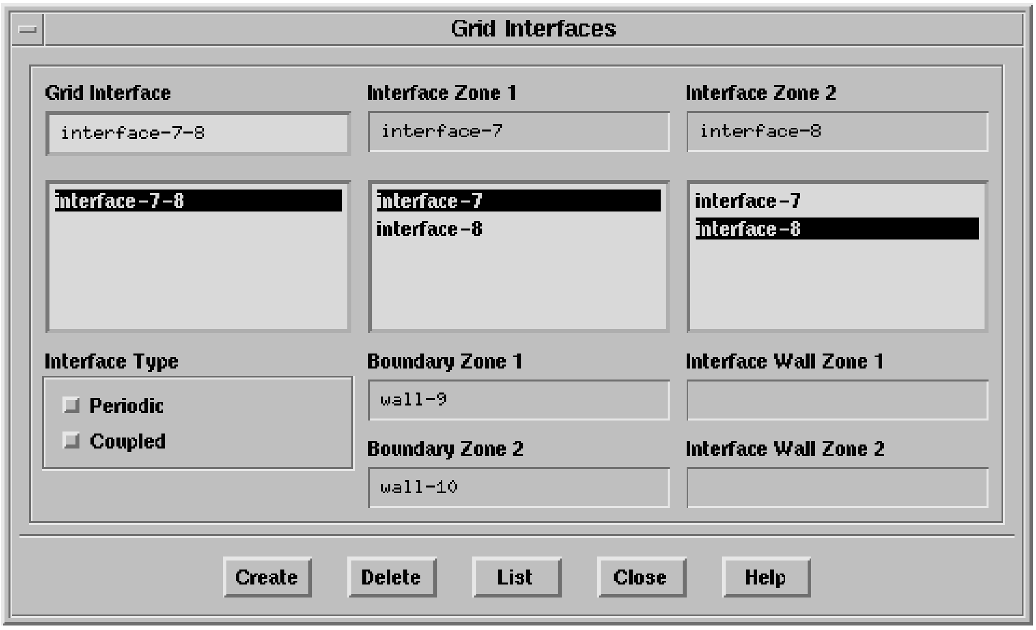


图 7-4 网格交界面

在 Grid Interfaces 中输入交界面的名称，在 Interface Zone 1 和 Interface Zone 2 中指定交界面两侧的流动区域，同时在 Interface Type 中设定交界面类型（周期性问题选 Periodic，

如果交界面位于固体区域和流体区域之间则选 Coupled），最后点击 Create 按钮完成网格交界面的创建。

因为滑移网格技术通常用于非定常问题的计算，所以计算过程中应该注意保存每个时间段的算例文件和数据文件。如果计算的是周期性流动问题，比如旋转机械问题，则在计算开始的时候可以采用大的时间步长以缩短开始阶段的不稳定过程，在计算稳定下来后再减小时间步长以保证时间精度，同时每隔一个周期观察一下流场变量的变化。如果变化逐渐缩小，说明计算是稳定的。在变化量小于 5%时，可以认为计算已经收敛。

在采用二阶时间精度进行计算的过程中，开始阶段可以采用大的时间步长，但是在随后减小时间步长的过程中，每次减小不要超过 20%。在计算的最后阶段，最好不要再改变时间步长，因为那样会严重影响计算的时间精度。

在后处理过程中，速度场的显示在缺省设置中显示的是绝对速度，也可以根据需要将其改为显示相对速度。

7.3.5 动网格技术

动网格技术用于计算运动边界问题。在前面的计算中，通常计算域的边界都是静止的，或做刚体运动的，而动网格技术则可以计算边界发生形变的问题。边界的形变过程可以是已知的，也可以是取决于内部流场变化的。在计算之前首先要给定体网格的初始定义。在边界发生形变后，其内部网格的重新划分是在 FLUENT 内部自动完成的。而边界的形变过程即可以用边界函数来定义，也可以用 UDF 函数来定义。

如果计算域中同时存在运动区域和静止区域，因此在初始网格中，内部网格面或区域需要被归入其中一个类别，同时在运动过程中发生形变的部分也可以单独分区。区与区之间即可以采用正则网格，也可以采用非正则网格，还可以用滑移网格技术连接各网格区域。

基本思路

动网格的计算方法有 3 种，即弹性光顺法（spring-base smoothing）、动态层技术（dynamic layering）和局部网格重划法（local remeshing）。

弹性光顺法将网格系统看作由节点之间用弹簧相互连接的网络系统，初始网格就是系统保持平衡的弹簧网络系统。任意一个网格节点的位移都会导致与之相连接的弹簧中产生弹性力，进而导致临近网格点上的力平衡被打破。由此波及出去，经过反复迭代最终整个网格系统达到新的平衡时，就可以得到一个变形后的、新的网格系统。

动态层技术是根据边界的移动量动态地增加或减少边界上网格层的技术，因此动态层技术适用于六面体网格、楔形网格等可以在边界上分层的网格系统。动态层技术在边界上假定一个优化的网格层高度，在边界发生移动、变形时，如果临近边界的一层网格的高度同优化高度相比大到一定程度时，就在边界面与相邻网格层之间增加一层网格。相反，如果边界向内移动，临近网格被压缩到一定程度时，临近一层网格又会被删除。用这种办法保持边界上的网格保持一定的密度。

局部网格重划法是对弹性光顺法的补充。在网格系统用三角形或四面体网格组成时，如果边界的移动和变形过大，可能导致局部网格发生严重畸变，甚至出现体积为负的情况。在这种情况下，一个简单的处理方法就是去掉由原来网格系统经过弹性光顺得到的新网格，在原来的位置上重新划分网格，这就是局部网格重划法的基本思路。

基本设置

（1）在求解器面板中选择非定常计算。
（2）在边界条件面板中设置动壁面条件。
（3）在动网格模型中设置相关参数：
Define->Dynamics Mesh->Parameters...

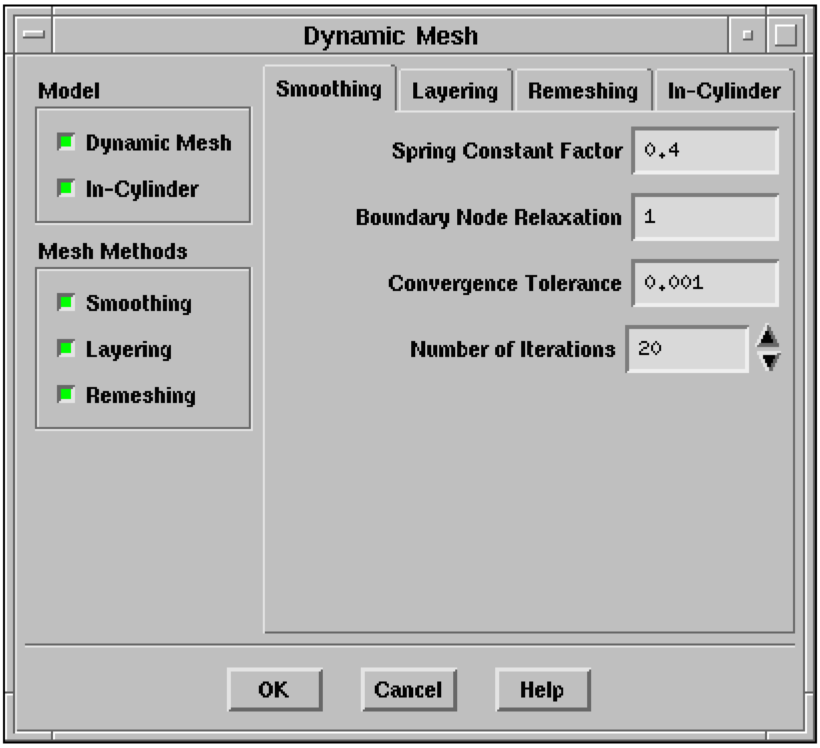


图 7-5 动网格参数设置

汽缸内流动是一类典型的边界移动问题，因此被 FLUENT 单独作为一类问题进行处理。在动网格参数设置中，首先需要选择的就是 Model 中的是否采用动网格（Dynamic Mesh），及所计算的模型是否属于汽缸内的流动（In-Cylinder）。其次就是选择网格的划分方法，包括前面所述的 3 种方法。

在选定网格划分方法后需要在右边的参数标签中选择需要设定的参数，即设定弹性光顺法（Smoothing）、动态层技术（Layering）和局部网格重划法（Remeshing）中需要设定的参数。如果所计算的问题是汽缸内流动（In-Cylinder），还需要为这种流动进行专门的设

定。

在弹性光顺法的参数中，弹性因子（Spring Constant Factor）是介于 0 到 1 之间的一个常数，其中 0 表示弹簧没有阻尼作用。边界节点松弛（Boundary Node Relaxation）则是一个类似于亚松弛因子的参数，每次迭代后新的网格坐标值都等于原坐标加上边界节点松弛与坐标增量的积，缺省设置为 1。其余两个参数则是网格迭代计算的控制参数，一个是容差（Convergence Tolerance），另一个是迭代次数（Number of Iterations）。

在动态层技术中，可以设置的参数包括常数高度（Constant Height），或常数比（Constant Ratio）。分裂因子（Split Factor）和消灭因子（Collapse Factor）是决定在网格变形到什么程度时，需要增加或消除网格层的控制因子。

局部网格重划中最大网格畸变（Maximum Cell Skewness）、最小网格体积（Minimum Cell Volume）和最大网格体积（Maximum Cell Volume）是决定哪些网格需要被集中起来重新划分的依据。畸变率大的，体积过小的和过大的都可以集中起来，进行重新划分。重新划分间隔（Size Remesh Interval）决定新网格的大小，“必须改进的畸变（Must Improve Skewness）”选项则决定是否需要在网格达到要求后再停止重划。

汽缸内流动则通过设定曲柄轴速度（Crank Shaft Speed）、曲柄轴起始角度（Starting Crank Angle）和曲柄轴周期（Crank Period）等定义曲柄周期运动。还有一个需要设定的参数是曲柄角度步长（Crank Angle Step Size），这个参数用于确定曲柄角度的计算步长。FLUENT 同时还提供一个内建的函数用于计算曲柄位置，即活塞顶端位置（Piston Stroke）和连接杆长度（Connecting Rod Length）。

（4）定义动态区域的运动：

Define->Dynamic Mesh->Zones...

首先选择区域名称（Zone Names），然后选择运动类型（Type），包括静止（Stationary）、刚体运动（Rigid Body）、变形（Deforming）和用户定义（User-Defined）等几种。然后设定运动相关属性，即在标签 Motion Attributes 下面根据计算模型设定重心位置等参数。

（5）保存算例文件和数据文件。

（6）预览网格设置：

Solve->Mesh Motion...

首先保存算例文件。然后设定时间步数量（Number of Time Steps）和时间步长（Time Step Size）。当前时间显示在当前网格时间（Current Mesh Time）中。如果计算的是汽缸内流动，则时间步长是用曲柄角度步长（Crank Angle Step Size）和曲柄速度（Crank Shaft Speed）计算出来的。如果要显示动态网格，打开显示网格（Display Grid）选项，并设定显示频率（Display Frequency）。如果需要将显示内容保存在文件中，可以打开保存硬拷贝（Save Hardcopy）选项。点击 Preview 按钮开始预览。

（7）如果求解的是汽缸内的流动问题，则定义与计算相关的事件参数：

Define->Dynamic Mesh->Events...

这个选项仅在选择使用了 In-Clinder（汽缸内流动）模型时可以被打开。首先增加 Number of Events（事件数量），然后在 On（打开）下面点击相应的开关，并输入事件名称，和曲柄角度，最后点击 Define（定义）按钮进入 Define Event（定义事件）面板。根据具体

情况进行适当的设置后，点击 OK 按钮保存并推出。为了确定设置效果，可以在 Dynamic Mesh Events（动态网格事件）面板中点击下面的 Preview（预览）按钮进行观察。如有错误，可以重新进行设置。

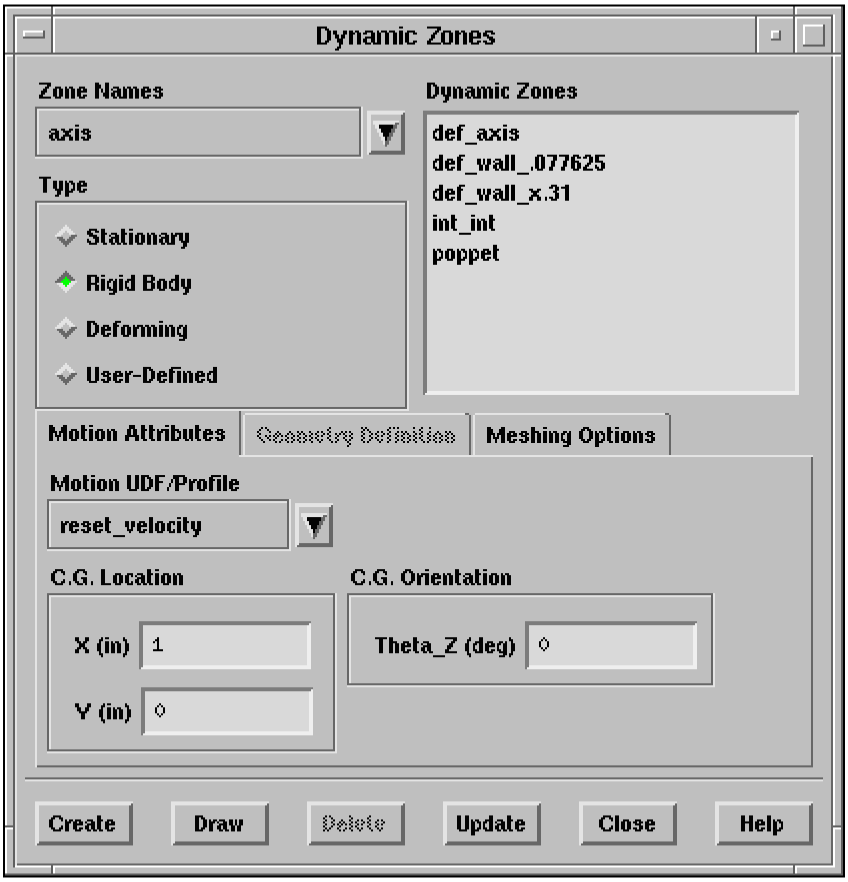


图 7-6 动态区域参数设置

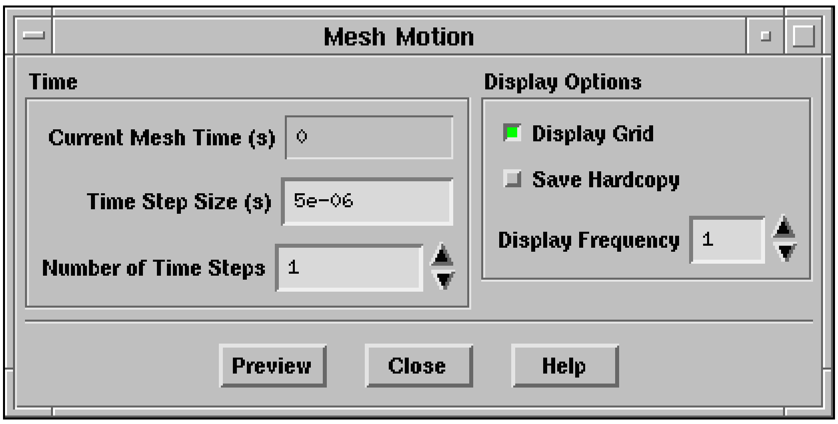


图 7-7 网格运动

除了在 Dynamic Mesh Events 面板中使用 Preview 按钮，还可以用下面的菜单操作显示汽缸内活塞和阀门的运动：

Display->IC Zone Motion...

（8）设置自动保存，以便隔一定的计算步保存一次算例文件和数据文件。

（9）计算中可以启动动画录制过程。

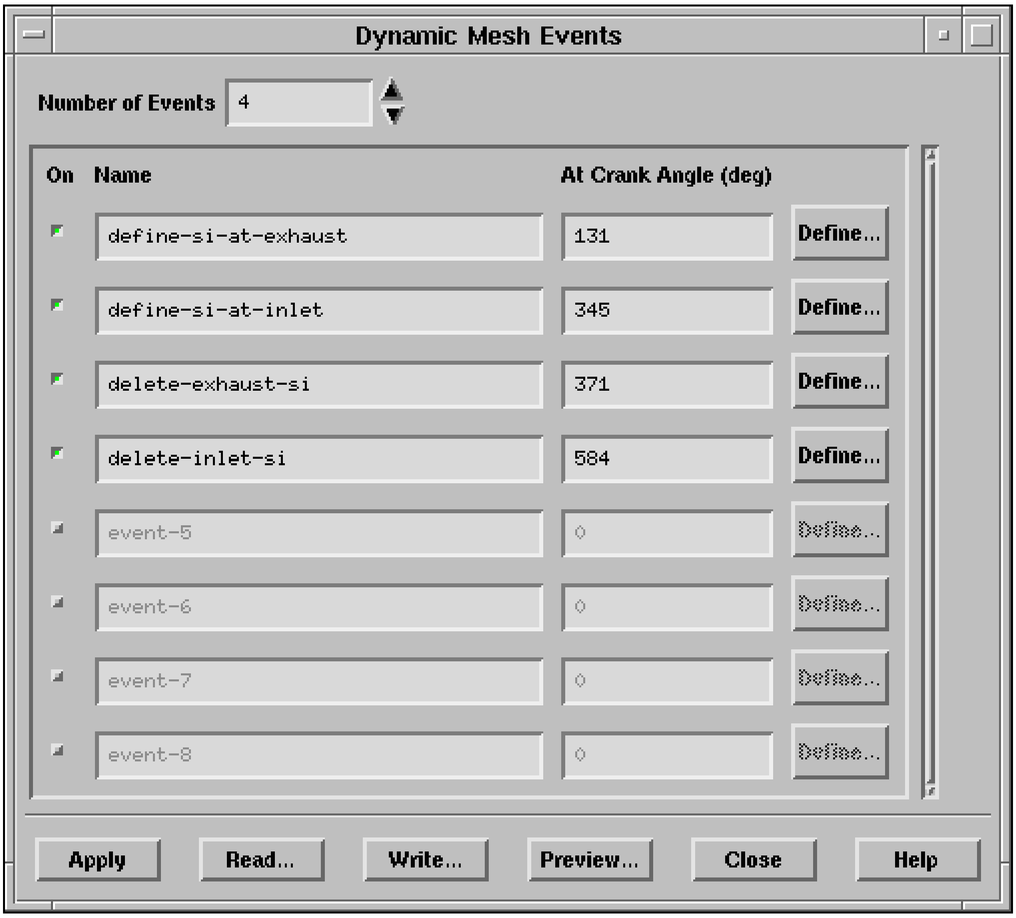


图 7-8 动态网格事件面板

7.4 化学反应模型

FLUENT 可以计算化学组分输运和化学反应过程。化学反应过程从类型上看可以分为气相反应、物面反应和弥散相粒子表面的反应。在气相反应中可以采用的计算方法又包括：

- 有限速率化学反应模型。
- 非预混燃烧模型。
- 预混燃烧模型。

- 部分预混燃烧模型。
- PDF 输运燃烧模型。

选择反应模型的一般方法为：

（1）对于涉及到化学反应、混合和输运过程的，或者反应发生在壁面或弥散相粒子表面的，采用有限速率化学反应模型。

（2）在计算湍流火焰扩散问题时，如果燃料和氧化剂是从多个不同入口进入流场，则可以使用非预混模型。

（3）对于所有反应物都均匀混合在一起的情况，可以使用预混燃烧模型。

（4）在火焰为预混火焰，而等价比（equivalence ratio）变化的问题，应该采用部分预混燃烧模型。

（5）如果在湍流火焰的计算中必须考虑有限速率化学反应的话，可以使用层流火苗模型，或组合物 PDF 输运模型。

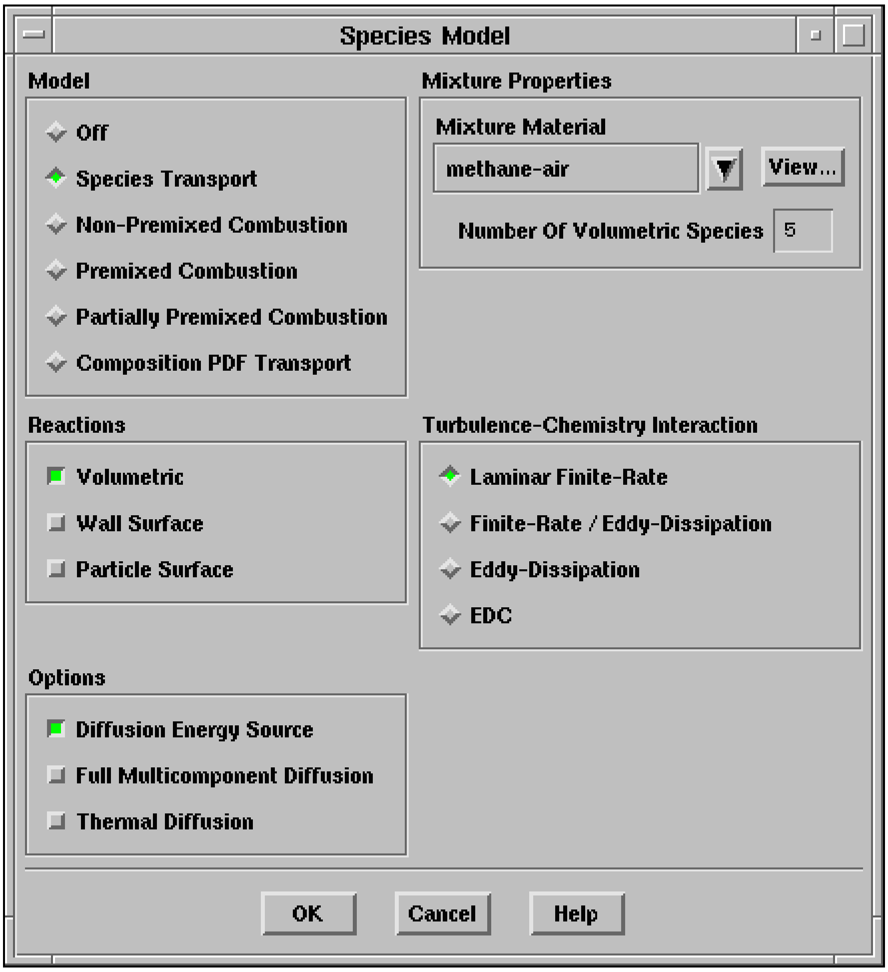


图 7-9 组元模型面板

7.4.1 气相反应

本小节只介绍有限速率化学反应，与燃烧和 PDF 有关的内容请参阅后面的 7.5 和 7.6 两节。有限速率化学反应模型的设置过程如下：

首先启动 Species Model（组元模型）面板，如图 7-9 所示：

Define->Modals->Species...

（1）在 Model（模型）中选择 Species Transport（组元输运）。

（2）在 Reactions（反应）中选择 Volumetric Reactions（体积反应）。

（3）在 Mixture Material（混合物材料）中选择所计算问题中涉及到的反应物，则 Number of Volumetric Species（体积组元数量）中自动显示混合物中的组元数量。

（4）在 Turbulence-Chemistry Interaction（湍流与化学反应相干模型）中根据需要选择相应的模型。

（5）如果选择了 EDC，则可以进一步修改 Volume Fraction Constant（体积浓度常数）和 Time Scale Constant（时间尺度常数）

（6）如果想完整计算多组分的扩散或热扩散，就选中 Full Multicomponent Diffusion（完整多组分扩散）和 Thermal Diffusion（热扩散）选项。

在上面的设置过程中，如果需要查看混合物中组分和化学反应的相关设置，可以在 Species（组元）面板中，点击 Mixture Material（混合物材料）右边的 View（观看）按钮。如果计算中用到的混合物模型是一种新的混合物，则需要在 Material（材料）面板中创建混合物，然后再将新定义的混合物选作计算用的混合物。

混合物的定义过程包含组分选取、反应模型设定、反应机制设定等几个步骤，下面逐一介绍。

组分选取

首先进入 Materials（材料）面板：

Define->Materials...

在 Materials（材料）面板上，先在 Name（名称）中为新的混合物确定一个名称，然后在 Material Type（材料类型）里选择 mixture（混合物）。如果有与目标相近的混合物模型，可以在下面的 Mixture Material（混合物材料）中选择一样，比如 methane-air（甲烷－空气），然后在下面 Properties（性能）中做详细设置，即按顺序设置组元、反应类型、反应机制等等：

（1）点击 Mixture Species（混合物组元）右边的 Edit（编辑）按钮进入 Species（组元）面板，如图 7-11 所示。在 Mixture（混合物）下面有 4 个框，即 Available Materials（可用材料），Selected Species（已选组元），Selected Site Species（已选吸收组元）和 Seleted Solid Species（已选固体组元）。

Available Materials（可用材料）是指材料数据库中可供选用的材料；Selected Species（已选组元）是指当前混合物中已经选中的组元；Selected Site Species（已选吸收组元）是

指在存在物面反应的计算中，气相混合物中即将通过反应被物面吸收的组元；Selected Solid Species（已选固体组元）是指物面反应计算中将从物面进入气流的组元。显然，如果不存在物面反应，则不用考虑后面两个方框中的内容，整个设置过程会大大简化。

组元设置的中心任务是选择混合物组元，即设定 Selected Species（已选组元）的内容。在 Materials（材料）面板中，点击 Database...（数据库）按钮打开 Database Materials（数据库中材料）面板，拷贝所需的组元后，再回到 Species（组元）面板。在拷贝之前，需要确认 Material Type（材料类型）必须是 fluid（流体）。这里不必担心如何添加物面反应中参与反应的固体组元，因为实际上这些组元也会出现在 fluid（流体）的列表中。

在 Selected Species（已选组元）中，最后一项必须是质量浓度最大的一个组元。如果最后一项不是质量浓度最大的一项，可以先将质量浓度最大的组元从方框中删除，然后再重新添加进来，以保证这个组元处于方框的底部。添加和删除操作是通过选择相应的组元，然后点击 Add（添加）和 Remove（删除）按钮实现的。

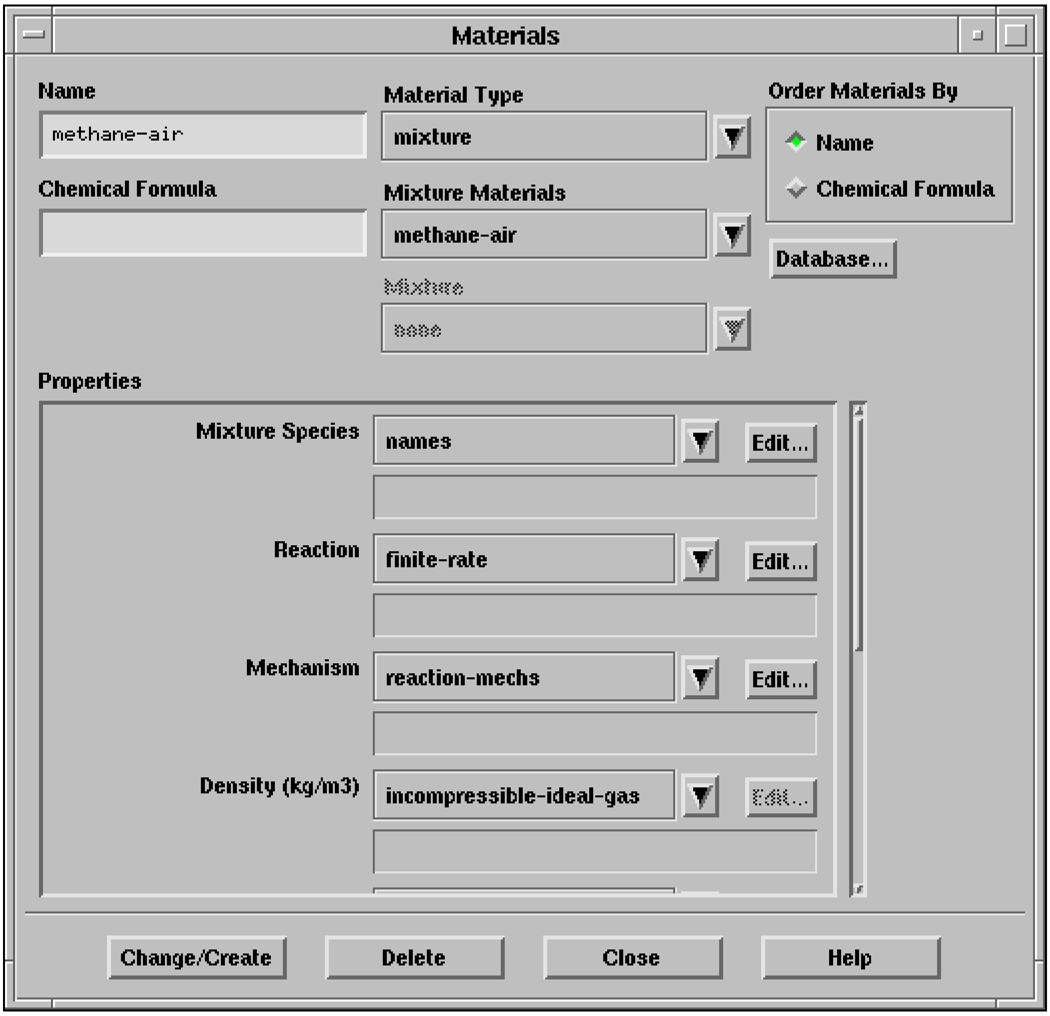


图 7-10 材料面板

化学反应设定

组元设置完成后，就可以开始设置组元间的化学反应。在 Materials（材料）面板中，Reactions（反应）下拉列表中显示的反应类型取决于 Species Modal（组元模型）面板中 Turbulence-Chemistry Interaction（湍流－化学反应相干）模型的设置——如果设置的是 Laminar Finite-Rate（层流有限速率）模型或 EDC 模型，则反应类型显示为 finite-rate（有限速率）；如果设置的是 Eddy-Dissipation（涡耗散）模型，则反应类型显示为 eddy-dissipation（涡耗散）；如果设置的是 Finite-Rate/Eddy-Dissipation（有限速率/涡耗散）模型，则反应类型显示为 finite-rate/eddy-dissipation（有限速率/涡耗散）。点击右端的 Edit（编辑）按钮进入 Reactions（反应）面板（如图 7-13 所示），在 Reactions（反应）面板中完成对化学反应模型的设置。



图 7-11 Species（组元）面板

化学反应的设置主要包括下列几项内容：

1）在 Total Number of Reactions（总的反应数量）中设定总的反应数量；

2）在 Reaction Name（反应名称）中指定反应名称；

3）在 Reaction ID（反应编号）中指定每个反应的编号；

4）在 Reaction Type（反应类型）中指定反应类型，即指定反应类型为 Volumetric（体

积）、Wall Surface（物面）或 Particle Surface（颗粒表面）；

5）在 Number of Reactants（反应物数量）和 Number of Products（生成物数量）中指定反应的反应物数量和生成物数量。

然后在 Species（组元）下拉菜单中选择反应物和生成物，并在 Stoich. Coefficient（反应系数）中设定组元在指定反应方程中的系数。在 Rate Exponent（速率指数）中设定组元的速率指数，即组元生成速率方程中指定反应方程中摩尔浓度项的指数。

6）如果在 Species Modal（组元模型）面板的 Turbulence-Chemistry Interaction（湍流－反应相干模型）中选择的是 laminar finite-rate（层流有限速率）、finite-rate/eddy-dissipation（有限速率/涡扩散）或 EDC 模型时，则需要根据反应模型设置 Arrhenius Rate（Arrhenius 速率）下面的选项。由于篇幅所限，这里不再详述。

7）根据实际反应过程，确定是否选择 Include Backward Reaction（包含逆向反应）选项。

8）如果使用 eddy-dissipation（涡扩散）或 finite-rate/eddy-dissipation（有限速率/涡扩散）模型，则还需要设定 Mixing Rate（混合速率）。

9）重复 2）～8）步直到设置完所有反应，然后点击 OK 按钮完成全部设定过程。

在区域上定义反应机制

FLUENT 中提到的“反应机制（reaction mechanisms）”指的是局限在特定区域中的化学反应。“反应机制”中涉及的反应是前面设定的化学反应的子集。在 Materials（材料）面板中，点击 Mechanisms（机制）旁边的 Edit（编辑）按钮，可以打开 Reaction Mechanisms（反应机制）面板，如图 7-14 所示。具体步骤如下：

1、在 Number of Mechanisms（反应机制数量）中设定反应的数量。

2、设定 Mechanisms ID（反应机制编号）。

3、设定 Name（名称）。

4、在 Reaction Type（反应类型）中设定反应类型。在反应类型确定后，属于这个类型的反应就会出现在 Reactions（反应）列表中。

5、选定反应机制中包含的反应。如果选定的反应类型是 Wall Surface（壁面反应）并且其中包含吸收反应，则还需要对吸收反应进行专门的设定，即设定 Number of Sites（吸收反应数量）、Site Name（吸收反应名称）、Site Density（吸收密度），点击 Define（定义）按钮还可以进一步选定吸收反应中的被吸收组元和被吸收组元的 Initial Site Coverage（初始吸收覆盖率）等参数。

设定混合物性质和组元性质

在上述设置结束后，可以分别设定混合物的物理性质，即密度、粘度、热导率和质量扩散系数等参数。同时还需要设定组元的理化性质，包括分子量、生成焓、粘度、热导率、比热和参考温度等参数。

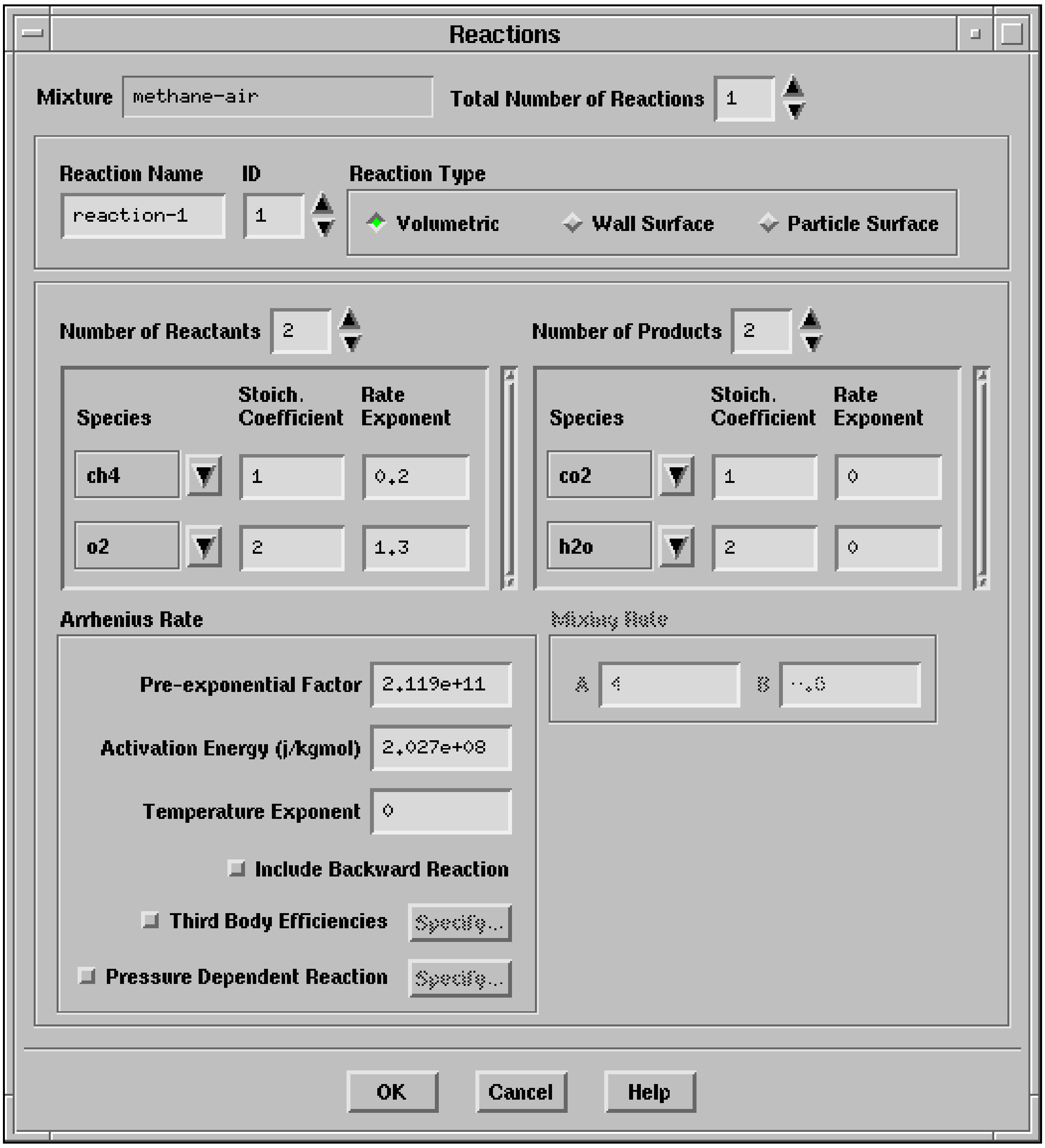


图 7-12 Reactions（反应）面板

边界条件

在有化学反应的计算中，需要在边界上设定组元的质量浓度。在没有壁面反应的时候，通常将壁面上的质量浓度梯度设为零。另外，还可以在 Fluid（流体）面板中为某种组元设置源项。FLUENT 还可以直接从 CHEMKIN 中输入反应模型，菜单操作为：

File->Import->CHEMKIN Mechanism...

指定了相关文件的路径并为反应命名后，就可以直接从 CHEMKIN 中读入反应模型了。

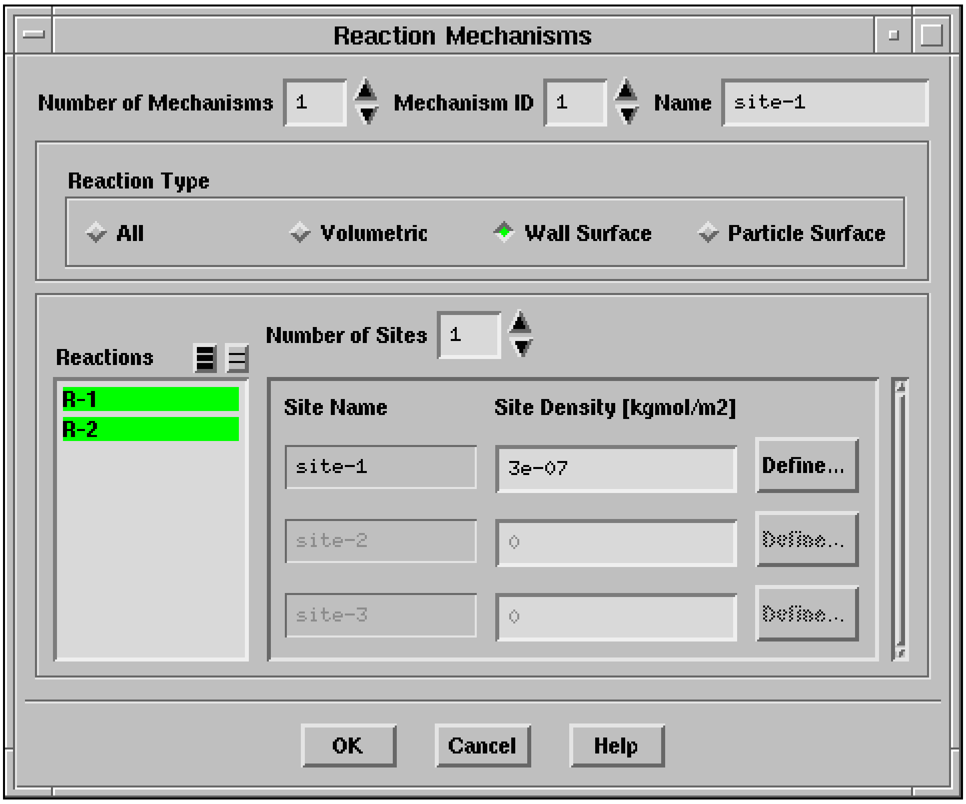


图 7-13 Reaction Mechanisms（反应机制）面板

7.4.2 壁面反应

壁面反应是发生在壁面及附近区域的化学反应。同前面的气相反应相比，明显的一个不同是，壁面反应中有些组元不是通过参与其他反应转变为其他组元，而是通过反应沉积在物面上，因此在组元方程中对源项的物理模型是不同的。同时壁面反应是以单位面积组元的生成率作为基本度量标准的，而在气相反应中是以单位体积来进行度量的。虽然有这些不同，在具体操作上是基本类似的：

（1）首先设定组元模型。在 Species Model（组元模型）面板上进行下列操作：

1）在 Model（模型）下面选择 Species Transport（组元输运）选项。

2）在 Reactions（反应）下面选择 Volumetric（体积）和 Wall Surface（壁面）选项。

3）在 Mixture Properties（混合物性质）下选择 Mixture Material（混合物材料）。

4）如果需要模拟壁面反应传热过程，则开启 Heat of Surface Reactions（表面反应热）。

5）如果需要模拟传质过程，则开启 Mass Deposition Source（质量沉积源项）。

6）如果计算使用的是分离算法，并且不希望在能量计算中考虑组元扩散效应，则关闭 Diffusion Energy Source（扩散能量源项）。

7）如果需要全面模拟多组分扩散火热扩散过程，开启 Full Multicomponent Diffusion（全

面多组分扩散）或 Thermal Diffusion（热扩散）选项。

（2）定义混合物性质。如同在气相反应中那样设定混合物中包含的组元、反应和组元性质等参数。壁面反应中可能被壁面吸收的组元也会出现在 Fluid Material（流体材料）列表中，因此将流体类型设定为 fluid（流体）就可以在列表中找到所有参与反应的组元。

（3）检查混合物中每个组元的性质，如果性质参数不符合反应模型的要求，可以进行组元性质的重新设定。

（4）设定组元边界条件。如果需要考虑壁面反应对壁面的影响，可以在 Wall（壁面）面板中 Species（组元）框内，开启 Surface Reactions（壁面反应）选项。在需要特殊处理的壁面附近设定壁面区附近的“反应机制”。

7.4.3 粒子表面反应

模拟弥散相颗粒表面发生的化学反应是燃烧计算中必不可少的内容。

计算模型设置

在计算中的设置过程如下：

（1）在 Species Model（组元模型）面板中的 Reactions（反应）框下面，开启 Particle Surface（粒子表面反应）选项。

（2）指定参与粒子表面反应的所有组元，包括参与沉积过程的组元。

（3）设定表面反应，即在 Reactions（反应）面板中，Reaction Type（反应类型）下设定反应类型为 Particle Surface（粒子表面反应），并指定下列参数：

1）Diffusion Limited Species（扩散限制组元），即在粒子表面和气流间梯度最大的组元。在只存在一种气相反应物时，没有必要设置这个参数。

2）Catalyst Species（催化组元）。在反应物中不包含固体颗粒时，固体颗粒是被作为催化剂加以定义的，即指定固体颗粒为催化组元。

3）Diffusion Rate Constant（扩散率常数）。

4）Effectiveness Factor（效率因子）。

采用多重表面反应模型

如果在颗粒表面反应中采用多重表面反应模型，其设置过程如下：

（1）启用所有弥散相模型选项。

（2）定义初始条件。

（3）定义边界条件。

（4）定义材料性质。

（5）如果参与表面反应的组元数大于 1，则回到 Set Injection Properties（设置引射性质）面板，或 Set Multiple Injection Properties（设置多重引射性质）面板，定义表面反应组元的质量浓度。

（6）设置求解参数并开始求解。

（7）检查结果。

上面各项的设置过程细节请参考本章后面关于弥散相计算的相关内容。

7.4.4 组元输运

除了进行气相反应计算、颗粒表面反应计算外，FLUENT 还可以进行组元混合问题计算，即在不发生化学反应的情况下计算组元之间的相互参混过程。这一计算过程可以看作是对气相反应计算的简化，基本步骤如下，详细过程请参考前面的反应流计算：

（1）在 Species Model（组元模型）面板中打开 Species Transport（组元输运）选项。

（2）如果需要计算多重扩散和热扩散，则开启 Full Multicomponent Diffusion（全面多重扩散）选项或 Thermal Diffusion（热扩散）选项。

（3）定义混合物及组元性质，即定义混合物中的组元构成和相应的物理性质。

（4）设置组元边界条件。

（5）设置计算条件。

（6）开始计算。

7.5　燃烧模型

燃烧计算分为非预混燃烧、预混燃烧和部分预混燃烧三种，本小节将针对三种算法分别进行介绍。

7.5.1 非预混燃烧模型

非预混燃烧适用于燃料和氧化剂分别来自不同入口的情况，也就是说燃料和氧化剂在燃烧前没有进行过混合，这就是所谓"非预混"的含义。

非预混燃烧计算中不使用有限速率化学反应模型，而是用统一的混合物浓度作为未知变量进行求解，因此计算中无需计算代表组元生成或消失的源项，计算速度比有限速率化学反应模型要快。但是非预混燃烧计算需要流场满足一定的条件，即流场必须为湍流，化学反应过程的驰豫时间非常短，燃料、氧化剂必须来自不同的入口等等。在燃料入口和氧化剂入口之外，非预混燃烧允许存在第三个流动入口，这个入口可以是燃料、氧化剂，也可以是不参与燃烧反应的第三种流体的入口。

非预混燃烧计算使用的化学反应模型包括火焰层近似（flame sheet approximation）、平衡流计算和层流火苗（flamelet）模型三种模型。火焰层近似模型假设燃料和氧化剂在相遇后立刻燃烧完毕，即反应速度为无穷大，其好处是计算速度快，缺点是计算误差较大，特别是对于局部热量的计算可能超过实际值。平衡流计算是用吉布斯自由能极小化的方法求

解组元浓度场，这种方法的好处是既避免了求解有限速率化学反应模型，同时又能够比较精确地获得组元浓度场。层流火苗模型则将湍流火焰燃烧看作由多个层流区装配而成，而在各层流子区中可以采用真实反应模型，从而大大提高计算精度。

非预混燃烧计算中湍流计算采用的是时均化 NS 方程，湍流与化学反应的相干过程用概率密度函数（PDF）逼近。计算过程中组元的化学性质用 FLUENT 提供的预处理程序 prePDF 进行计算处理。计算中采用的化学反应模型可以是前面所述三种模型中的一种。计算结束后将计算结果保存在查阅（look-up）表格中，FLUENT 在计算非预混燃烧时则直接从查阅表格中调用数据。

PrePDF 操作流程

prePDF 的操作过程如下：

（1）启动 prePDF。

（2）分配内存。

分配内存的方式，就是在 prePDF 中指定组元、混合物浓度平均点、混合物浓度增量点、焓的取值点、层流子区的最大数值，确定了这些量的最大值后，系统分配给 prePDF 的最大内存就基本确定了。需要注意的是，如果需要改变这些值，必须重新启动 prePDF。分配内存在 Memory Allocation（内存分配）面板中进行。在 prePDF 中，Memory Allocation（内存分配）面板的启动方式为：

Setup->Memory Allocation...

（3）初始化问题定义。

内存分配完毕后，需要对化学反应系统类型和化学反应模型进行定义，即确定是否存在燃料入口和氧化剂入口之外的第三个入口、部分预混模型选项、绝热或非绝热模型选项、选择化学反应模型、PDF 函数形式、用经验公式定义燃料入口和第三个入口流动组成等等。这些选择过程在 Define Case（定义算例）面板中完成，这个面板的启动方法为：

Setup->Case...

（4）定义化学组元。

模型选定后再指定反应系统中的组元。这些组元必须是 prePDF 数据库中收录的组元，其定义在 Define Species（定义组元）面板中完成，Define Species（定义组元）面板的启动方法为：

Setup->Species->Define...

在面板中指定组元总数，每个组元的编号和名称，然后从数据库中依次选择即可。

（5）定义燃料入口、氧化剂入口的组分构成。

在设定完系统中的组元后，还需要指定各种组元在燃料入口、氧化剂入口中的浓度。这项操作在 Composition（构成）面板中进行，其启动方法为：

Setup->Species->Composition...

在设置过程中，首先指定是燃料入口还是氧化剂入口，然后指定组元编号，并指定浓度定义形式（摩尔浓度或质量浓度），最后指定浓度值即可。

（6）定义操作条件。

操作条件在 Operating Conditions（操作条件）面板中完成，其启动方法为：

Setup->Operating Conditions...

根据实际情况依次设定相关参数即可。

（7）定义化学反应参数。

在采用火焰层近似模型或部分预混模型计算化学反应时，需要设定化学反应模型中的相关参数。PrePDF 中化学反应系数的输入方法与 7.4 节中所说的略有区别，比如在下面的反应中：

$$CH_4+2O_2->CO_2+2H_2O$$

CH_4的计量系数是 1，O_2的系数是 2，CO_2的系数是-1，H_2O 的系数是-2，即需要在生成物的系数前面加上负号。计量系数是在 Stoichiometric Coefficients（化学计量系数）面板中完成的。这个面板的启动方法为：

Setup->Species->Stoichiometry...

在这个面板中首先指定参与反应组元的总数，然后分别指定各种组元的计量系数，并如前所述，用正负号区别反应物和生成物。

（8）定义创建查阅表格的参数。

PrePDF 的计算结果存放在查阅表格中，控制查阅表格的参数在 Solution Parameters（求解参数）面板中设定。这个面板的启动方法为：

Setup->Solution Parameters...

在这个面板中设定非绝热模型中查阅表格中焓的计算点数（Enthalpy Points），燃料混合物浓度的计算点数（Fuel Mixture Fraction Points），燃料可燃极限（Fuel Rich Flamability Limit）等参数。

（9）保存所有输入值。

操作方法为：

File->Write->Input...

（10）计算查阅表格。

保存输入后，开始计算：

Calculate->PDF Table

（11）保存查阅表格。

计算结束后保存查阅表格：

File->Write->PDF...

查阅表格可以保存为有格式文件（ASCII 或文本文件）和无格式文件（二进制文件），可以用于 FLUENT4 或 FLUENT6 等不同的求解器。在保存查阅表格时，要注意上述区别。

PrePDF 中带有图形显示、报告等后处理功能，可以用于检验查阅表格的计算结果，这里不再详述。

非预混燃烧模型设置

非预混燃烧包括单一混合物浓度计算（Single-Mixture-Fraction Approach）和双混合物浓度计算（Two-Mixture-Fraction Approach）。大致步骤如下：

（1）启动 FLUENT 并读入网格文件。

（2）选择使用非预混燃烧模型。

首先在 Viscous Model（粘性计算模型）面板中选用湍流计算。如果计算采用的是非绝热模型，则还需要在计算中加入热交换计算等内容。

然后在 Species Model（组元模型）面板中选择 Non-Premixed Combustion（非预混燃烧），点击 OK 按钮后，系统会自动打开一个文件选择对话框，从中可以选择由 prePDF 创建的 PDF 文件。如果需要在计算中引入压缩性效应的话，可以返回 Species Model（组元模型）面板，在左下角点击选中 Compressibility Effects（压缩性效应）。

（3）定义边界条件。

非预混燃烧边界条件的特殊之处就在于混合物浓度在边界上的定义。通常平均燃料混合物的浓度在燃料入口，浓度值为 1，在氧化剂入口则为 0；氧化剂则正好相反，即在燃料入口为 0，在氧化剂入口为 1，也就是假设燃料入口和氧化剂入口流入的是纯净的燃料或氧化剂。

（4）定义物理性质。

在指定使用非预混模型后，流体介质既被自动设定为 pdf-mixture（pdf 混合物），其成分已经在 prePDF 中指定，并且在 FLUENT 中无法更改。在 FLUENT 的 Materials（材料）面板中只能修改粘度、热导率等输运性质。

（5）开始流场计算。

流场计算开始前，先在 Species Model（组元模型）中确认各项参数设置无误，特别是需要采用的 PDF 函数形式等项与预期目的相符后，就可以进行初始化，然后开始迭代计算了。如果计算不收敛，则可以尝试适当调整亚松弛因子重新开始计算。

7.5.2 预混燃烧模型

预混燃烧即燃烧前燃料和氧化剂已经充分混合了的燃烧。预混燃烧的火焰传播速度取决于层流火焰传播速度和湍流对层流火焰的相干作用。湍流中的旋涡结构可以使火焰锋面发生变形、起皱，并进而影响火焰传播速度。在预混燃烧中，燃烧反应的反应物和燃烧的生成物被火焰区截然分开。

在 FLUENT 中预混燃烧必须使用分离算法，并且只能用于湍流、亚音速流动计算。预混燃烧模型不能与污染物（NO_x）模型同时使用，但是可以与部分预混模型同时使用。预混燃烧不能用于模拟带化学反应的弥散相粒子，但是可以模拟带惰性粒子的流动计算。

预混燃烧模型计算的设置和求解过程如下：

（1）启用预混湍流燃烧模型并设置相关参数。

Define -> Models -> Species...

在 Species Model（组元面板）中选择 Premixed Combustion（预混燃烧）模型，并在下面选择 Adiabatic（绝热）或 Non-Adiabatic（非绝热）。

（2）定义未燃烧材料的物理性质。

Define -> Materials...

在这里定义与预混燃烧有关的混合物性质，包括 Laminar Flame Speed（层流火焰传播速度）、Critical Rate of Strain（临界应变率）、Heat of Combustion（燃烧放热）等等。

（3）设置过程变量 c 在流场入口和出口处的值。

Define -> Boundary Conditions...

在流场的入口和出口处设置过程变量 c，对于未燃烧混合物 c=0，对于已燃混合物 c=1。

（4）初始化过程变量。

Solve -> Initialize -> Patch....

可以在全流场将 c 值设为 1（已燃），然后让未燃混合物（c=0）从流场入口进入流场，并将火焰推回到火焰稳定器处。另一个比较好的选择是在驻燃区上游用补丁（patch）方式将 c 值设为 0（未燃），而在下游设为 1（已燃）。

（5）求解流场并进行后处理。

7.5.3 部分预混燃烧模型

部分预混燃烧模型是非预混燃烧模型和预混燃烧模型的综合体，计算中火焰锋面的位置用过程变量 c 计算，在锋面后面（c=1）是已燃的混合物，锋面前面（c=0）则是未燃的混合物。部分预混燃烧模型适用于混合物混合不充分的燃烧计算。

部分预混燃烧的设置和求解流程：

（1）在使用 prePDF 创建 PDF 查阅表格时，基本流程与非预混燃烧相同，区别在于在 Define Case（定义算例）面板上选择的选项是 Partially Premixed Model（部分预混模型）。

（2）在 FLUENT 中读入网格文件，并设置好计算中要用到的其他模型，比如湍流模型、辐射模型等等。

（3）在 Species Model（组元模型）面板中激活 Partially Premixed Combustion（部分预混燃烧）模型。如果有必要，可以在面板中修改 Model Constants（模型常数）。在点击 OK 按钮后，文件选择窗口会自动打开，在这里选择由 prePDF 创建的包含查阅表格的 PDF 文件。

（4）在 Materials（材料）面板中，定义计算域中未燃材料的物理性质。

（5）作为边界条件，在流场入口和出口处设置过程变量 c 和平均混合物浓度及其增量的值。

（6）初始化过程变量。

Solve -> Initialize -> Patch...

（7）开始计算，计算结束后对感兴趣的变量进行后处理。

7.6　PDF 输运模型

PDF 输运模型用有限速率化学反应模型计算湍流火焰。PDF 输运模型用概率密度函数法模拟湍流流动，可以模拟火焰的点火过程和火焰的消失过程，但是 PDF 输运模型消耗的系统资源很大，因此计算中不应该使用太多的网格点，最好将计算限于二维情况。

PDF 输运模型计算仅能用分离算法进行，并且不能用于模拟变化的热传导过程。用于化学反应计算的 ISAT 表不能以并行方式进行输入输出操作。

7.6.1 设置和求解过程

PDF 输运模型的设置和计算过程如下：

（1）在 CHEMKIN Mechanism（CHEMKIN 反应机制）面板中，读入 CHEMKIN 文件中的气相反应模型。CHEMKIN Mechanism（CHEMKIN 反应机制）面板的启动方法为：

File -> Import -> CHEMKIN Mechanism...

（2）启用湍流模型。在 Viscous Model（粘性模型）面板中，选择计算中使用的湍流模型。Viscous Model（粘性模型）面板的启动方法为：

Define -> Models -> Viscous...

（3）启用 PDF 输运模型并设置相关参数。首先启动 Species Model（组元模型）面板：

Define -> Models -> Species...

然后选择 Composition PDF Transport（组合物 PDF 输运）模型。选中后，Species Model（组元模型）面板会扩展显示与 PDF 模型相关的输入选项。

在 Species Model（组元模型）上的 Reactions（反应）选项组中，选择 Volumetric（体积），点击 Integration Parameters（积分参数）打开 Integration Parameter（积分参数）面板。在这个面板上设置 ODE 积分容限和 ISAT 积分参数。缺省设置的积分参数可以满足大部分计算的要求，因此如果对积分过程没有特殊的要求，可以保持缺省值不变。

最后在 Species Model（组元模型）面板上，选择粒子混合模型，即选择 Modified Curl（改进的卷曲）模型，或 IEM（Interaction by Exchange with the Mean，与平均流进行交换的相互干扰）模型。

（4）在 Materials（材料）面板中检查材料性质，并在 Reactions（反应）面板中检查反应参数。

（5）设置操作条件（Operating Conditions）和边界条件（Boundary Conditions）。

（6）检查求解器设置。

在 Solution Controls（求解过程控制）面板中的 PDF Transport Parameters（PDF 输运参数）下面，设置 Particles per cell（每个网格中的粒子数量）和 Local Time Stepping（当地时间步长）参数。

（7）初始化流场，在需要的时候可以用补丁（patch）的方式在流场中定义一个高温

区用于点火。

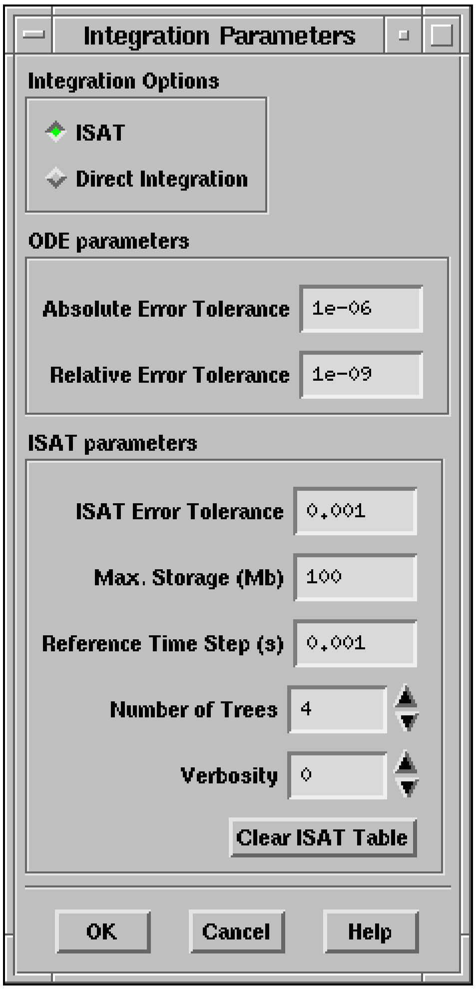


图 7-14 Integration Parameters（积分参数）面板

（8）设置求解过程监视器，然后开始迭代求解。

在低速流中，燃烧通过密度与流场流动耦合起来。蒙特卡罗（Monte Carlo）PDF 输运算法给予密度场随机扰动，并进而形成对整个流场的扰动。在定常流计算中，随机扰动对稳定性的破坏作用可以通过预迭代过程的平均作用予以减弱。在计算中，流场经常在温度场和组元浓度场收敛之前就已经“收敛”。为了解决这个问题，可以适当调整流场收敛判据，同时在 Flux Reports（通量报告）面板中确认 Total Heat Transfer Rate（总的热交换速率）已经达到平衡。同时建议在出口边界上监视温度场和组元浓度场的变化，保证温度和组元浓度在出口边界上也已经达到稳定状态。

缺省设置中，FLUENT 每进行一次有限体积迭代就进行一次 PDF 计算。在实际计算过程中，有限体积迭代和 PDF 计算的步数都可以进行调整。

增加 Iterations in Time Average（时间平均计算的迭代步数），流场的扰动会被光滑掉。缺省设置的 50 步迭代可以适用于多数算例。如果需要逐步减少残差，可以将 Time Average Increment（时间平均增量）设为 0 到 1 之间的一个数（推荐设置为 0.2），后续计算将根据这项设置增加时间平均计算的迭代步数。

（9）对计算结果进行后处理。

在计算结束后，除了能对静温、组元质量浓度等变量进行文字和图像的后处理外，还可以跟踪 PDF 计算过程中的粒子，即在 Particle Tracks（粒子跟踪）面板中，选择 Track PDF Transport Particles（跟踪 PDF 输运粒子），然后点击 Display（显示）按钮，就可以观察 PDF 输运粒子的运动轨迹。

7.6.2 ISAT 算法

化学反应流动是计算流体力学中的难题之一，特别是在涉及真实化学反应流动模型时就更是如此。真实化学反应模型中涉及的组元和反应数量都非常多，其中有些反应速度很快，有些反应速度很慢，为了准确计算快速的反应，时间步长就必须取得很小，而用很小的时间步长去计算速度慢的反应则要耗费大量的计算时间，这就是化学反应计算中所谓的“刚性问题”。

用各组元的质量浓度和流场温度、压强作为分量，可以定义“粒子构成矢量”。化学反应计算可以看作是从粒子构成矢量从初始值演化到稳态值的过程。FLUENT 计算化学反应流动的方法有直接积分法（Direct Integration, DI）和 ISAT（In-Situ Adaptive Tabulation，在计算过程中不断调整的表格法）方法。

直接积分法是以粒子构成矢量的初始值为起点，通过对化学反应源项的积分获得每个时间步长后的值。ISAT 法则将粒子构成矢量的可能取值制作成一个分支图，在计算中尽量不用积分，而直接从图中查询相关矢量值，因此可以省去积分占用的计算时间，从而大大加速反应计算。

ISAT 算法的计算逻辑是：

（1）对于一个粒子构成矢量的初始值，首先在分支图中横向查找与之相近的矢量，并以最接近的矢量为起点进行线性插值运算，所得结果与相近矢量相比较，如果能够满足误差要求，则采用这个值作为粒子构成矢量在一个时间步后的新值。

（2）如果比较结果不满足误差要求，则需要用直接积分的方式获得一个新的矢量值。将这个矢量值与前面的插值结果（而不是相近矢量）相比较，得到的误差值如果能够满足原来的误差要求，则改变误差容限，以便将粒子构成矢量的初始值纳入相近矢量的误差容限范围。在后续计算中如果再遇到类似情况时，就可以直接使用插值进行计算了。

（3）在直接积分结果与插值结果相比较，仍然不能满足误差容限的话，则在分支图中增加一个节点，并定义这个节点上的梯度、误差容限。在后续计算中进行横向搜索时，这个新的节点被作为搜索对象之一，在粒子构成矢量的初始值与这个节点的情况接近时，就采用这个节点的定义进行计算。

从上面的叙述中可见，ISAT 分支图在计算过程中是不断增加变化的。随着其增加和变化，计算中的直接积分运算次数会逐渐减少。理想情况下，当这个分支图不再变化时，所有化学反应计算都可以用查阅的方式完成，则计算工作量会大大降低。

在 ISAT Parameters（ISAT 参数）面板中对 Verbosity（冗长度）进行设置，就可以对 ISAT 的计算过程进行监视。将 Verbosity（冗长度）设为 1，则 FLUENT 将与 ISAT 有关的信息写入文件 isat_stats.dat，这些信息包括查阅操作总数、取回数据总数、误差容限调整总数、节点增加总数、节点替换总数等等。Verbosity（冗长度）设为 2 是为熟悉 ISATAB 的专家级用户设置的，这里不再详述。在大多数情况下，将 Verbosity（冗长度）设为 1 就可以了。

在并行计算中，每个计算节点的 ISAT 表格是单独建立的，在 Verbosity 文件中，都包含一个计算节点的编号以示区别。

为了提高 ISAT 算法在实际计算中的使用效率，需要注意以下几点：

（1）分支图上有些节点对应的是初始流场中的矢量值，在后续的计算中是没有用的，因此可以通过清理表格的方式，清除这些节点，提高计算效率。清除节点可以在 Integration Parameters（积分参数）面板中进行。先保存算例文件和数据文件，然后再重新启动，也可以达到清理的目的。

（2）在计算收敛后，可以将最终形成的 ISAT 表保存起来，在需要的时候再重新读入。保存的菜单操作是：

File -> Write -> ISAT Table...

读入 ISAT 表的菜单操作是：

File -> Read -> ISAT Table...

（3）在 ISAT 表被填满后，不能通过查阅操作完成的计算，将通过直接积分完成，因此在计算设置中应该尽量加大 ISAT 表的存储量。

（4）使用多个表格可以提高 ISAT 计算的效率。采用多个表格可以减小每个表格的存储量，因此可以减少搜索时间，同时小的 ISAT 表格还可以让内存较小的计算机进行比较复杂的化学反应计算。

（5）如果 ISAT 计算的误差容限设置的比较小，则计算达到收敛的时间会大大加长。可以考虑的一种计算策略是在计算的初期采用较大的误差容限，在计算收敛后再减小误差容限，这样可以兼顾精度和速度。

从化学反应的机理上进行分析，可知 ISAT 算法在做预混燃烧模型计算时速度比较快，因为预混燃烧中包含的状态空间（由粒子构成矢量组成的空间）比较小。而在非预混燃烧中，燃料和氧化剂质量浓度的变化范围都远大于预混燃烧模型，其状态空间也因此比较大，所以在非预混燃烧计算中，ISAT 的表现会略逊于预混燃烧计算。

7.7 弥散相模型

FLUENT 用弥散相模型计算散布在流场中的粒子的运动和轨迹，例如在油气混合汽中，空气是连续相，而散布在空气中的细小的油滴则是弥散相。连续相的计算可以用求解流场

控制方程的方式完成，而弥散相的运动和轨迹则需要用弥散相模型进行计算。弥散相模型实际上是连续相和弥散相物质相互作用的模型。在带有弥散相模型的计算过程中，通常是先计算连续相流场，再用流场变量通过弥散相模型计算弥散相粒子受到的作用力，并确定其运动轨迹。

弥散相计算是在拉格朗日观点下进行的，即在计算过程中是以单个粒子为对象进行计算的，而不象连续相计算那样是在欧拉观点下，以空间点为对象。比如在油气混合汽的计算中，作为连续相的空气，其计算结果是以空间点上的压强、温度、密度等变量分布为表现形式的，而作为弥散相的油滴，却是以某个油滴的受力、速度、轨迹作为表现形式的。关于欧拉观点和拉格朗日观点的区别和相互转换，可以参考流体力学中的相关内容。

FLUENT 在计算弥散相模型时可以计算的内容包括：

- 弥散相轨迹计算，可以考虑的因素包括弥散相惯性、气动阻力、重力，可以计算定常和非定常流动。
- 可以考虑湍流对弥散相运动的干扰作用。
- 计算中可以考虑弥散相的加热和冷却。
- 计算中可以考虑液态弥散相粒子的蒸发和沸腾过程。
- 可以计算燃烧的弥散相粒子运动，包括气化过程和煤粉燃烧过程。
- 计算中既可以将连续相与弥散相计算相互耦合，也可以分别计算。
- 可以考虑液滴的破裂和聚合过程。

因为弥散相模型计算中可以包括上述物理过程，所以可以计算的实际问题也非常广泛。

7.7.1 弥散相模型的使用限制

在粒子的体积密度小于 10～12％时可以使用弥散相模型进行计算。需要注意的是，体积密度小于 10～12％时，粒子的质量密度可能远远大于这个比例，甚至大于连续相的质量密度。

弥散相模型适用于计算有出口和入口的流动问题，即适用于弥散相粒子不是长时间地停留在计算域内，而是从入口处飞入，再从出口处飞出的问题。另外弥散相模型不能与质量流入口或压强降低条件配合使用，不能与适应性时间推进同时使用，同时弥散相模型中的粒子与连续相之间没有化学反应。在弥散相粒子是从一个表面进入流场时，不能使用动网格技术，因为弥散相粒子所在平面不能随动网格一起移动。

7.7.2 弥散相计算策略

在定常流计算时，弥散相的计算策略为：

（1）求解连续相流动。

（2）创建弥散相射流。

（3）在需要的时候，用耦合方式求解流场。

（4）用图形或文字报告的方式跟踪弥散相粒子。

在非定常流场计算时，弥散相的计算策略为：

（1）创建弥散相射流。

（2）初始化流场。

（3）通过定义迭代步数的方式，控制计算的时间推进过程，开始计算。

7.7.3 弥散相粒子分类

弥散相模型中粒子的分类决定了在计算过程中将进行哪些热力学计算。在 FLUENT 中，粒子被分为三类，即惰性粒子、液滴和燃烧粒子，其对应的热力学计算如下：

（1）惰性粒子：加热和冷却。

（2）液滴：加热、蒸发和沸腾。

（3）燃烧粒子：加热、挥发、增长和表面反应。

7.7.4 粒子与湍流的相互作用

弥散相粒子与湍流之间的相互作用可以用随机跟踪模型或云雾模型进行计算。随机跟踪模型，即随机漫步模型，可以计算湍流的脉动速度对弥散相粒子轨迹的影响。而云雾模型则将弥散相粒子看作围绕粒子运动轨迹做概率分布的集团，粒子的空间分布用高斯概率密度函数表示，粒子的发展变化过程也用概率表达。

7.7.5 引射类型

弥散相粒子通过引射方式进入流场，而引射方式又可以分为雾化器模型、液滴碰撞模型、喷雾破裂模型和动态阻力模型等 4 个类型。

（1）雾化器模型。

雾化器模型与其他引射类型的区别是，雾化器模型中引射进入流场的粒子是在空间和时间上呈随机分布的，而其他引射类型则是在计算开始的时候按照固定的轨迹进入流场的。

雾化器模型具体包括 5 个模型，即普通喷嘴雾化器、压力旋转雾化器、平面雾化器、空气助力式雾化器和激变（effervescent/flashing）雾化器模型。

- 普通喷嘴雾化器是指液态射流从喷嘴中喷出，直接进入流场，其雾化过程与喷嘴几何构型及射流在喷嘴中的形态有关——射流在喷嘴中可能出现单相流（即液态射流充满喷嘴）、空化流（即液态射流中出现空泡）和收缩流（即射流直径小于喷嘴直径，射流周围被气体环绕）等 3 种形态。
- 压力旋转雾化器。

这类雾化器中的射流在出口处形成伞形液面，在伞形液面的边缘处，液态射流破裂成

为小的液滴，形成油雾。

- 平面雾化器。

与压力旋转雾化器类似，平面雾化器中的射流从一个扁平的喷口流出，并形成一个很薄的液面，然后液面破裂，形成油雾。

- 空气助力式雾化器。

这种雾化器在射流中加入空气流，通过空气与液流的相互作用促进破裂过程的发生，从而达到强化雾化效果。

- 激变雾化器。

激变雾化器中的射流为热的易于发生汽化的射流。在射流从喷口喷出后，骤然发生相变，达到雾化效果。

（2）液滴碰撞模型。

气流中散布的液滴数量是十分巨大的，因此其碰撞次数也非常大。如果跟踪每个粒子的飞行过程，并计算粒子之间两两的碰撞，所需要的系统资源将是个天文数字，实际上是不可能的，也完全没有必要。因此在实际的计算中，液滴碰撞的计算中引入了"包（parcel）"的概念，即假定液滴的运动可以看作是一个个的液滴包组成的，每个液滴包中包含了大量的液滴，比如包含了 1000 个粒子。这样一来，计算工作量就可以大大降低。

FLUENT 中使用的碰撞模型是 O'Rourke 模型。这个模型同时假定只有处于同一个网格单元中的"包"才能发生碰撞。碰撞的结果可能是液滴之间发生合并，或者相互分开。

碰撞模型假定碰撞过程的时间尺度远远小于粒子运动的时间尺度。如果粒子运动的时间非常大，则结果会显示出对时间步长的依赖性。此时需要相应调整粒子的长度尺度。另外，这个模型更适用于低 Weber 数的碰撞，此时碰撞的结果是粒子反弹或合并。在 Weber 数大于 100 时，碰撞的结果是散乱的。

因为模型中假设碰撞只能发生在同一个网格中，所以有时候计算结果明显显示出对网格的依赖。在二维问题中可以用细分网格和增加液滴数量的办法解决这个问题。在三维计算中可以用极坐标网格建立以喷雾为中心的模型，既可以获得较好的解。

（3）喷雾破裂模型。

FLUENT 中有两种喷雾破裂模型，一种是 Taylor 类比破裂（TAB）模型和波动模型。TAB 模型适用于低 Weber 数流动，特别是喷雾低速喷射到标准大气中的流动问题。波动模型适用于 Weber 数大于 100 的情况，特别是燃油高速引射问题。

- TAB 模型

  TAB 模型将液滴的破裂过程与弹簧的压缩变形过程相类比，将表面张力类比于弹簧的回复力，液滴气动阻力类比于弹簧受到的外力，液滴的粘性力类比于弹簧的阻尼。在液滴发生变化时，液滴受到的阻力也随之发生变化。在变形达到一定程度时，比如新形成的小液滴中心之间的距离大于原来大液滴的直径时，就可以认为液滴破裂为两个小液滴。

- 波动模型

  Reitz 的波动模型认为液滴的破裂过程取决于气体和液滴之间的相对速度，即相对速度大，气体与液滴之间的作用力就大，液滴在作用力下就更容易破裂成细小的液滴。液

滴的大小是从射流的稳定性分析中得出的。

（4）动态阻力模型。

因为引射模型中需要考虑液滴的破裂过程，而液滴的破裂则直接源于气体施加在液滴上的阻力，所以精确的阻力计算对于液滴运动的计算是非常关键的。FLUENT 中用动态阻力模型计算液滴阻力，在这个模型中可以考虑因为液滴形变而导致的阻力变化，因此可以满足工程应用的需要。

动态阻力模型适用于大多数情况，并与喷雾破裂模型中的 TAB 模型和波动模型兼容。液滴在运动中的变化过程可以简化为从球形向碟形的演变，动态阻力模型则根据液滴变形的大小，假定液滴阻力为球形液滴阻力与碟形液滴阻力之间的线性插值。

7.7.6 在计算中加入弥散相模型

打开弥散相模型（Discret Phase Model）面板：

Define->Models->Discrete Phase...

在这个面板上可以设置的项目包括：

（1）弥散相与连续相的相互干扰选项。

弥散相与连续相之间的干扰包括动量干扰、质量干扰和热交换过程，选择这个选项则启动计算过程中的弥散相与连续相之间的干扰计算。选中 Interaction with Continuous Phase 选项既启动此计算过程。

下面的“每次弥散相迭代间隔的连续相计算迭代次数（Number of Continuous Phase Iterations per DPM Iteration）”输入栏中的数字是前后两次弥散相计算之间进行的连续相计算迭代步数。缺省设置为 10，即每进行 10 步连续相计算就做一次干扰计算。

（2）弥散相计算选项。

Options 选项组中的 6 个选项分别是在干扰计算中可以考虑的 6 种干扰因素，可以根据需要选择加入：

- 如果需要加入辐射热交换，即选中 Particle Radiation Interaction 选项。
- 如果需要考虑热浮力对粒子轨迹的影响，则选中 Thermophoretic Force。
- 如果需要用耦合方式计算粒子的热和质量方程，则选中 Coupled Heat-Mass Solution 选项。在缺省设置中，这两个方程是用分离求解方式求解的。
- 在层流计算中，如果需要考虑布朗运动的影响，则选中 Brownian Motion 选项。同时应该在阻力算法（Drag Law）列表中选用 Stokes-Cunningham 阻力算法，并使用 Cunningham Correction 修正项。
- 如果需要考虑剪切力对粒子轨迹的影响，则选中 Saffman Lift Force 选项。
- 如果需要考虑粒子对墙壁的腐蚀和增厚作用，则选中 Erosion/Accretion 选项。同时需要设定粒子的入射角（Impact Angle）、粒子直径（Diameter Function）和速度指数函数（Velocity Exponent Function），这里假定已经为壁面选择了固体材料。

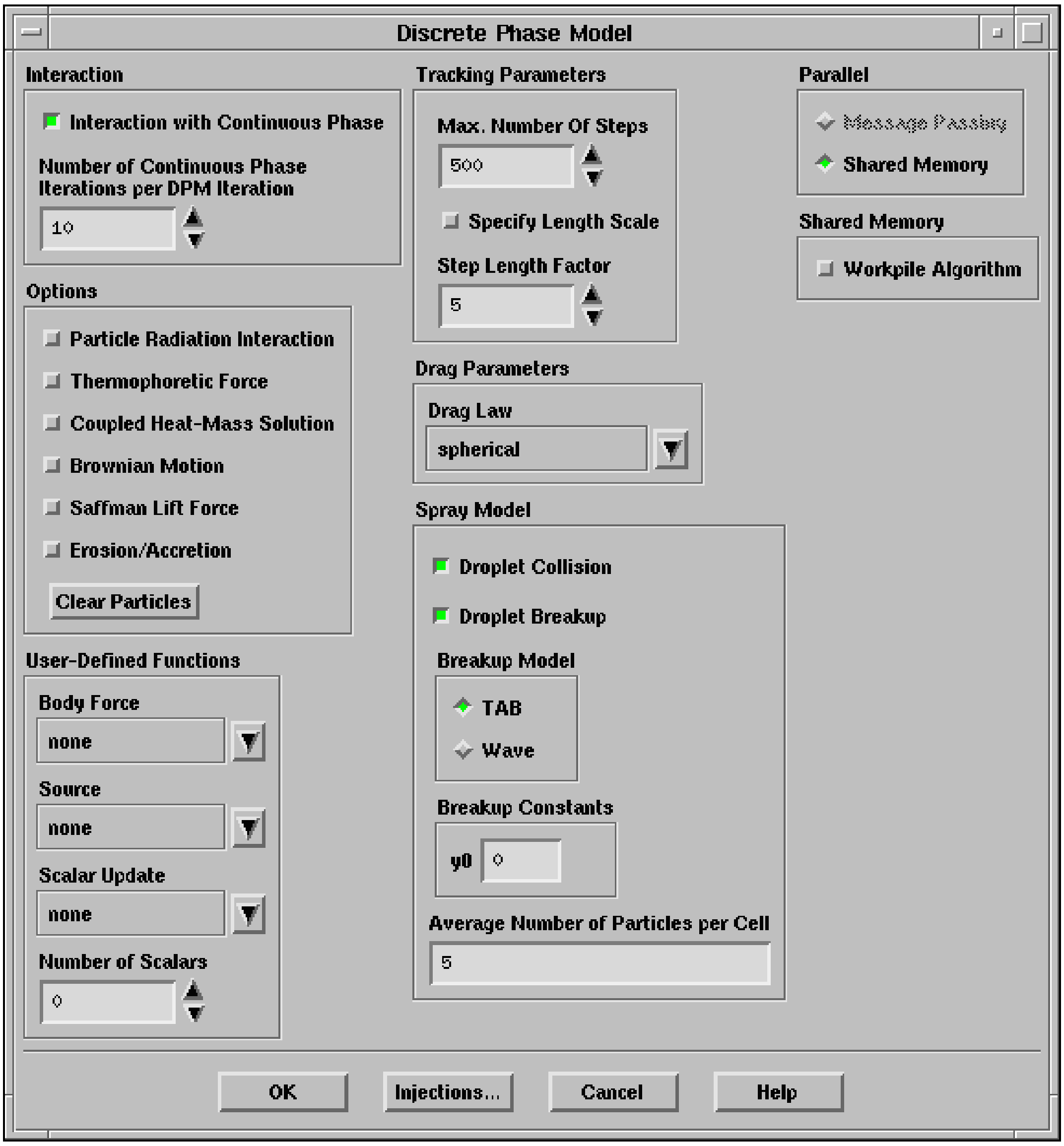


图 7-15 弥散相模型面板

（3）阻力算法。

在 Drag Law 列表中可供选择的阻力计算方法包括球形（Spherical）算法、非球形（non-spherical）算法、Stokes-Cunningham 算法和高马赫数（high-Mach-number）算法。

（4）UDF 函数。

在 User-Defined Functions 下面，可以选择的选项有彻体力（Body Force）、源项（Source）和变量更新（Scalar Update）等，在考虑对壁面的腐蚀和增厚作用时，还有腐蚀/增厚（Erosion/Accretion）选项。

（5）喷雾模型。

在 Spray Model 下面设置喷雾模型，包括液滴碰撞（Droplet Collision）和液滴破裂（Droplet Breakup）选项，同时还可以选择 TAB 模型或波动（Wave）模型为破裂模型，并设置破裂常数（Breakup Constants）和每个单元中的平均粒子数（Average Number of Particles per Cell）。

（6）在弥散相面板中部顶端是追踪参数（Tracking Parameters）设置，包括跟踪计算的最大步数（Max. Number of Steps）和步长因子（Step Length Factor）。最大步数的缺省值为 500，在计算中需要调整这个数以便能够正确跟踪粒子飞行轨迹。如果发现粒子轨迹在飞行中途停止不前，说明最大步数的值太小，需要加大步数；如果粒子轨迹能够顺利到达壁面或计算域的出口，则说明最大步数的设置可以满足计算需要。

（7）并行计算。

在用并行版 FLUENT 进行计算时，需要确定具体采用何种并行算法。在 Parallel 下包含两个选项，一个是消息传递（Message Passing），即 MPI 算法，另一个是共享内存（Shared Memory）算法。详细信息可以参考并行计算部分的说明。

7.7.7 弥散相模型的初始化

打开 Injections 面板：

Define->Injections...

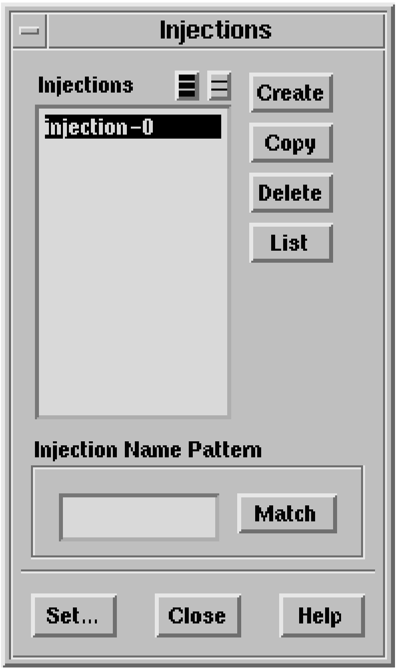


图 7-16 Injections（引射）面板

或在弥散相模型中点击 Injections...按钮，都可以进入 Injections 面板，如图 7-。再点击 Create 按钮进入设置引射模型属性（Set Injection Properties）面板，如图 7-　。

（1）首先在 Injection Name 里面给引射模型一个名称，也可以保留系统缺省设置的引射模型名称。

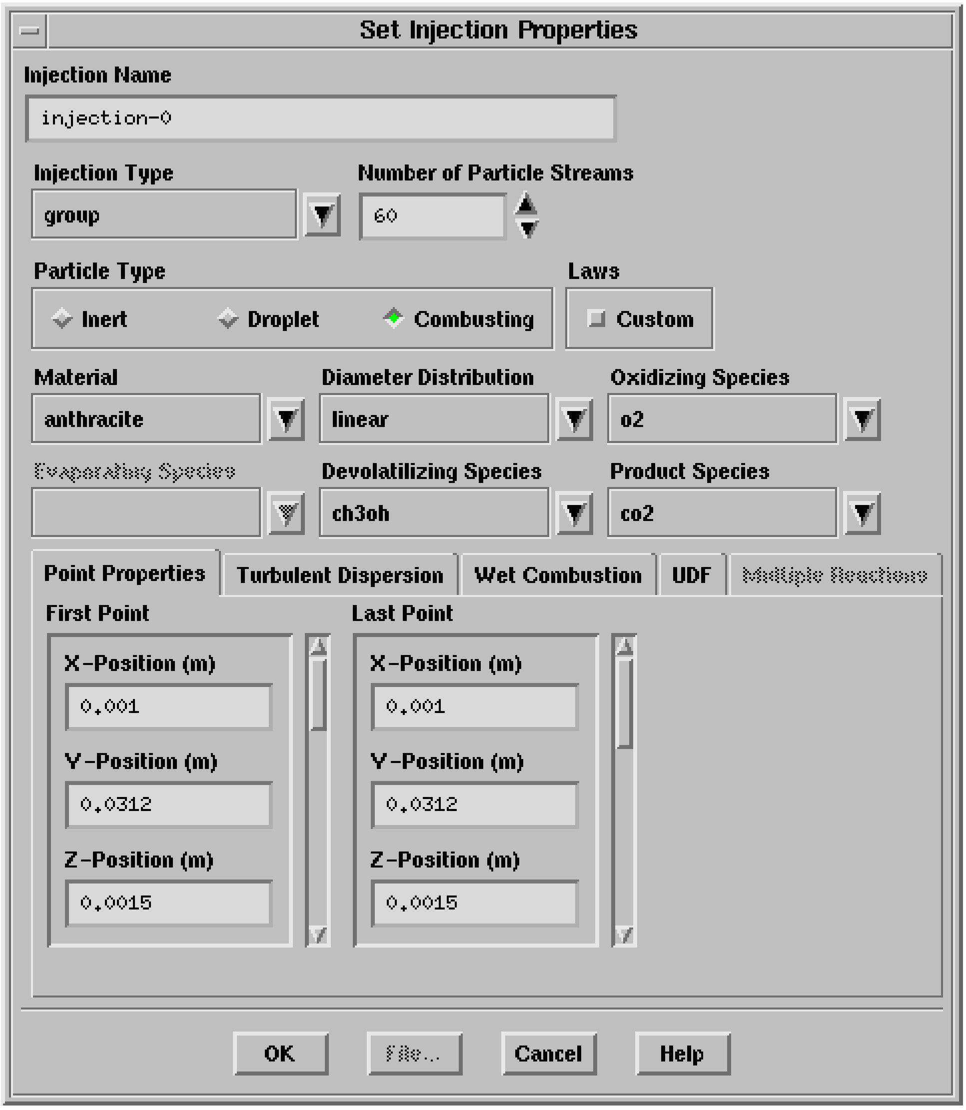


图 7-17 Set Injection Properties（设置引射性质）面板

（2）在引射类型（Injection Type）中选择射流类型，可供选择的引射类型包括：

- 单粒子（single）型。
- 粒子组（group）型。
- 三维空锥（hollow cone）型。

- 三维固体锥（solid cone）型。
- 面引射（surface）型。
- 普通出口雾化器（plain-orifice atomizer）型。
- 压力旋转雾化器（pressure-swirl atomizer）型。
- 平面扇形雾化器（flat-fan atomizer）型。
- 空气助力雾化器（air-blast atomizer）型。
- 激变雾化器（effervescent atomizer）型。
- 从文件读入（read from a file）。

雾化器模型前面已经有所介绍。非雾化器模型在被选中后，还需要为其设定初始位置、初始速度、粒子直径、温度、质量流量等相关参数。在前面几个模型无法描述计算中的引射类型时，还可以用外部编辑的文件对引射类型进行定义。

（3）在 Particle Type 中选择粒子的类型，即选择粒子为惰性粒子、液滴或燃烧粒子模型，三者的区别如前面所述。

（4）在 Material 中选择粒子成分，即确定粒子是什么物质构成的。这项选择将确定粒子的热力学属性，对于计算粒子挥发、合并、阻力等计算有直接影响。在粒子属性与系统缺省设置的有所不同时，可以对各参数的数值进行编辑。

（5）在确定了引射类型后，面板上的项目会发生相应的改变。比如在选定引射类型为粒子组(group)，粒子类型为燃烧粒子(combustion)以后，就需要设定组中的粒子数（Number of Particle Streams）、粒子直径分布形式（Diameter Distribution）、氧化剂成分（Oxidizing Species）、液化成分（Devolatilizing Species）和生成物（Product Species）等项目。这些都可以从计算的物理模型中直接得到。

（6）在 Point Properties 标签下设定引射点的初始位置、速度、质量流量、粒子直径等参数。

（7）在湍流散射（Turbulence Dispersion）标签中，可以选择湍流干扰模型，并设置相关参数，即选择随机跟踪（Stochatic Tracking Model）模型或云雾（Cloud Model）模型。这两种模型的区别请参考前面的介绍。

（8）在湿燃烧（Wet Combustion）标签中，设置燃烧模型参数。所谓湿燃烧，即燃烧过程中允许粒子存在挥发和沸腾现象。

（9）在 UDF 标签中，选择设置 UDF 函数相关参数。

7.7.8 设置弥散相计算的边界条件

弥散相边界条件分为反射（reflect）模型、捕捉（trap）模型、逃逸（escape）模型和壁面射流（wall jet）模型：

- 反射模型假定弥散相粒子在与壁面发生碰撞时粒子会反射回流场。在弥散相为固体颗粒时通常采用这种模型。
- 捕捉模型假定弥散相粒子与壁面碰撞时粒子被壁面捕捉，同时有部分质量以挥发

的形式重新进入流场。这种模型是液体颗粒在固体壁面上经常使用的模型之一。

- 逃逸模型假定弥散相粒子在与边界面碰撞后既离开流场，不再返回。这种模型通常用于流场的入口和出口处。
- 壁面射流模型将弥散相粒子与壁面的碰撞过程与射流与壁面碰撞过程相类比，从而得出碰撞结束后粒子的速度大小及方向。这个模型是在 FLUENT6.1 版本中增加的，适用于液体颗粒情况。

在设定边界条件时，可以发现面上的边界条件中包含一个 DPM 标签，DPM 是弥散相模型（Discrete Phase Model）的英文缩写，弥散相的边界条件就在这里设定。根据弥散相的性质在边界类型中进行选择即可。

在完成模型设置和边界条件设置后，需要重置弥散相模型：

Solve->Initialize->Reset DPM sources

到这里弥散相模型的设置完毕，下面就可以开始迭代计算了。

7.7.9 弥散相计算的后处理

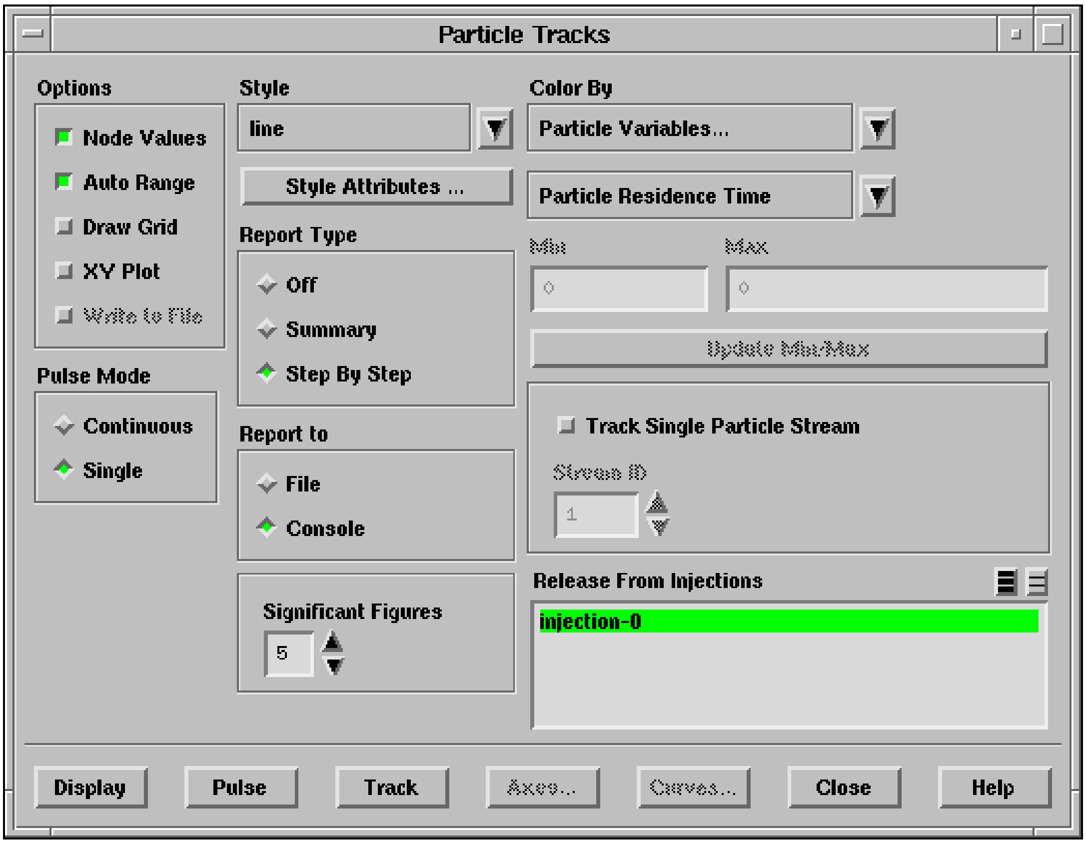


图 7-18　Particle Tracks（粒子跟踪）面板

弥散相计算结束后可以用图形和文本两种方式观察计算结果，主要内容包括：

（1）粒子轨迹的图形显示。首先打开粒子追踪（Particle Tracks）面板对显示过程进行设置：

Display->Particle Tracks...

- 在 Release From Injections 栏目中选择被跟踪的引射模型，然后对图形显示内容和形式进行设置后，点击 Display 按钮就可以在图形显示窗口中看到粒子迹线的显示。
- 粒子运动报告，即在控制台窗口中报告粒子总数及逃逸、捕捉、挥发、未完成计算的粒子数目。
- 每个迭代步后粒子位置、速度、温度和直径的报告，即报告单个粒子在每步迭代后的位置、速度、温度等信息。具体方法是：在粒子跟踪面板中，选择 Track Single Particle Stream 选项，并在报告类型（Report Type）下面选择逐步显示（Step By Step）选项，点击 Display 按钮就可以在控制台窗口中看到相关的报告了。也可以选择将报告记录以文件形式记录下来，只要在 Report to 中选择 File 即可。

（2）在 Report 菜单中的 Discrete Phase 中可以计算动量交换、热交换和质量交换信息，也可以对边界上的粒子轨迹进行采样，还可以计算壁面上的腐蚀和增厚速率等等。所有信息都在控制台窗口中显示。

7.8 多相流模型

本小节讲述 VOF 模型、混合物模型和欧拉模型的应用范围和使用方法。这三种模型都属于用欧拉观点处理多相流的计算方法，其中 VOF 模型适合于求解分层流和需要追踪自由表面的问题，比如水面的波动、容器内液体的填充等等；混合物模型和欧拉模型中则适合计算体积浓度大于 10％的流动问题，而上一节中提到的弥散相模型则适用于体积浓度小于 10％的流动问题。

混合物模型和欧拉模型的取舍方法为：

（1）如果弥散相粒子广泛地分布在流场各处，则采用混合物模型；如果弥散相粒子集中在流场的某个区域，则应该采用欧拉模型。

（2）在相间阻力定律适用于所计算的问题时，欧拉模型通常比混合物模型的计算精度更高。如果不知道相间阻力定律是否适用于所计算的问题，则可以选择使用混合物模型。

（3）混合物模型需要的系统资源较少，计算速度较快。欧拉模型比混合物模型精度高，但是计算时间长，稳定性也较差。

7.8.1 三种方法的限制条件

VOF 方法适于计算空气和水这样不能互相参混的流体流动，比如射流破裂过程、大型

气泡在液体中的运动、大坝溢流以及追踪气液自由表面的问题。计算中 VOF 法的限制是必须使用分离求解器，只有一种物相可以是可压缩流体，不能计算流动方向的周期性流动、组元混合与反应流动和无粘流动，不能与大涡模拟（LES）同时进行计算，不能采用二阶隐式时间推进格式，不能使用壁面的薄壳传导模型。VOF 方法通常用于模拟流体的非定常运动过程，但是也可以模拟定常运动问题。

混合物模型是一个简化的多相流模型，可以用于模拟存在相对运动速度的多相流问题，其应用范围包括粒子沉降过程、旋风分离器以及小体积比的气泡流动等问题。在使用上除了上面 VOF 法中提到的限制外，混合物模型还不能进行固化和熔化的计算。

欧拉模型可以模拟的多相流问题中，如果内存足够大的话，可以模拟任意数量物相的流动，但是在非常复杂的多相流问题中，计算的稳定性也是对求解的一个限制。在欧拉模型计算中，各种物相受到的背景压强是一样的，每种物相的动量方程和连续性方程都是单独求解的。对于流场中的固体颗粒，每种颗粒的温度都可以用代数方程来计算，而剪切和粘性则用动力学理论求出。针对不同的物相，可以采用不同的阻力系数计算函数。计算中可以针对每一种物相，或其混合物，采用 $k-\varepsilon$ 湍流模型进行湍流计算。

欧拉模型在计算上的限制是只能采用 $k-\varepsilon$ 模型进行湍流计算，只能对主要物相进行粒子跟踪，不能模拟流向周期流、可压流、无粘流、熔化和固化过程、组元输运和化学反应流动，计算中不能使用二阶隐式时间推进格式。

7.8.2 问题设置和计算流程

上述三种模型在 FLUENT 中的具体实现步骤如下：

（1）启用多相流模型（VOF 模型、混合物模型或欧拉模型），并参与流动的物相数量。在使用 VOF 模型时，同时设定 VOF 的组成，菜单操作为：

Define -> Models -> Multiphase...

在 Multiphase Model（多相流模型）面板上选择模型后，面板会展开出现与所选择的模型相关的选项。在选中 VOF 模型时，面板会展开让用户输入物相总数、计算格式和是否采用隐式彻体力计算；选中混合物模型后，需要输入物相总数、是否计算滑移速度、是否采用隐式彻体力计算、是否考虑空穴效应；选中欧拉模型后，需要输入物相总数，并确定是否考虑空穴效应。其中物相总数最多可以设为 20。

在混合物模型中，可以关闭滑移速度（Slip Velocity）选项，让所有物相以同一速度运动。

（2）从材料数据库中拷贝材料数据，菜单操作为：

Define -> Materials...

如果数据库中没有计算所需的材料数据，可以重新创建相关数据。在创建过程中，如果新材料是固体颗粒相，也要注意将其放在 fluid（流体）分类中。

如果在计算中需要考虑某个物相的压缩性，最好将这个物相设为主要物相（primary phase）。如果在某个边界上设定了总压（total pressure）值，则在同一个边界上设定的温度

值，对于可压缩物相就等于其总温，而对于不可压物相则等于其静温。对于质量流入口条件（mass flow inlet），需要定义每一个物相的质量流或质量通量。

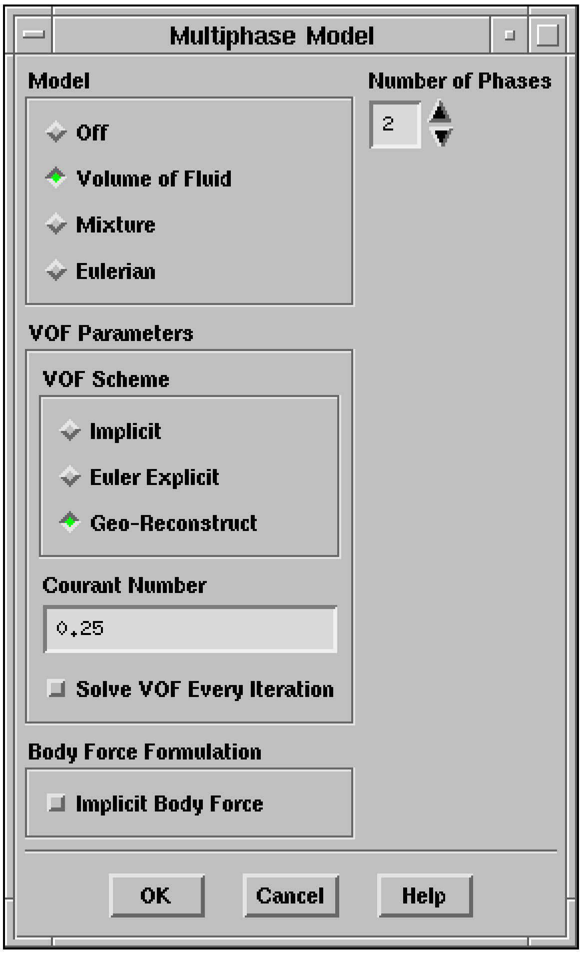


图 7-19　Multiphase Model（多相流模型）面板

（3）定义物相，以及相间作用。比如在使用 VOF 模型时定义表面张力，在使用混合物模型时定义滑移速度，在使用欧拉模型时定义阻力函数等等，菜单操作为：

Define -> Phases...

首先在 Phase（物相）面板中将计算中的各种物相定义为主要相（primary-phase）或次要相（secondary-phase），然后点击 Interaction...（干扰）按钮定义相间干扰。

7.8.3 VOF 模型

对于 VOF 模型来说，主要相设置的具体过程为：

1）在 Phase（物相）列表中选择 phase-1；

2）点击 Set（设置）按钮打开 Primary Phase（主要相）面板；

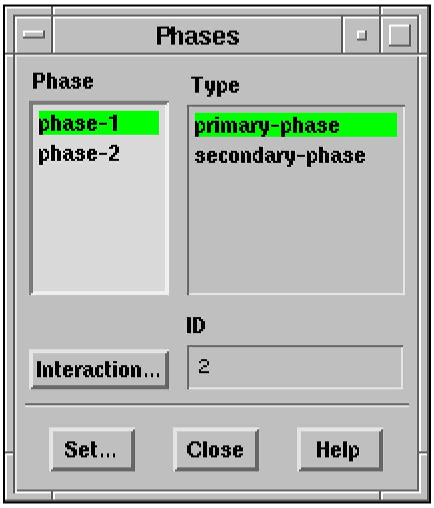


图 7-20 Phase（物相）面板

3）在 Primary Phase（主要相）面板中的 Name（名称）栏中设定物相的名称；
4）在 Phase Material（物相材料）下拉列表中指定物相所用的材料；
5）点击 Edit...（编辑）按钮对材料性质进行检查和编辑；
6）点击 Primary Phase（主要相）面板中的 OK 按钮完成设置。
次要相设置的具体过程为：
1）在 Phase（物相）列表中选择 phase-2；



图 7-21 Primary Phase（主要相）面板

2）点击 Set...（设置）按钮进入 Secondary Phase（次要相）面板；
3）在 Secondary Phase（次要相）面板中的 Name（名称）栏内设定物相名称；
4）在 Phase Material（物相材料）列表中选择物相材料；
5）检查、编辑材料属性；
6）在 Secondary Phase（次要相）面板中点击 OK 按钮完成设定过程。

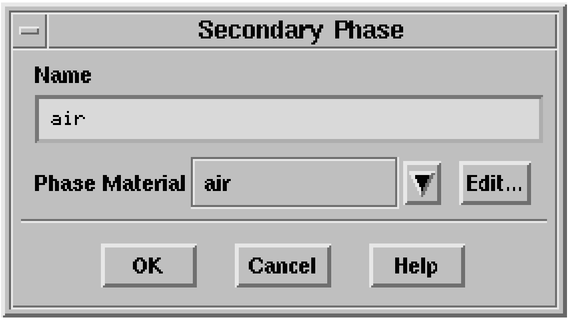


图 7-22 Secondary Phase（次要相）面板

设定完主要相和次要相后，点击 Interaction...（干扰）按钮设置相间干扰过程，对于 VOF 模型就是设置计算中的表面张力模型。

1）点击 Surface Tension（表面张力）标签；

2）如果需要考虑壁面粘性影响，点击打开左上角的 Wall Adhesion（壁面附着）选项。打开这个选项后，还需要设定各个壁面上的接触角；

3）在需要考虑表面张力影响的物相之间，比如空气和水之间，设定常数表面张力系数，或者设定温度相关的表面张力系数，也可以采用用户自定义形式的表面张力系数。

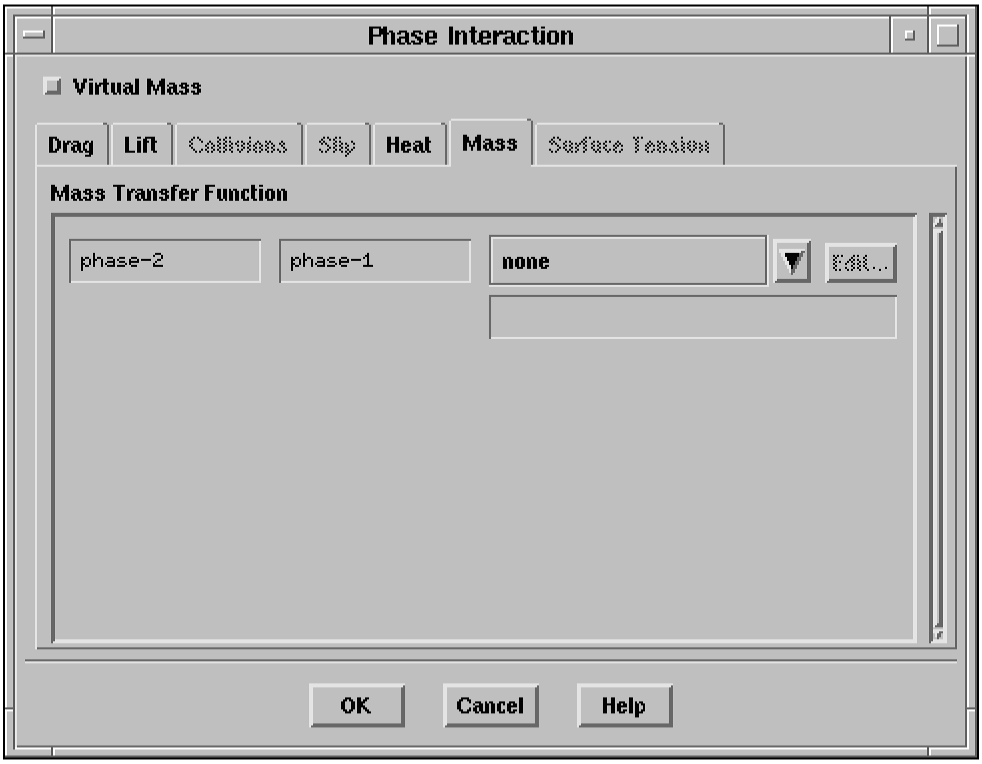


图 7-23 Phase Interaction（相间干扰）面板

在考虑表面张力作用的计算中，最好同时在 Multiphase Model（多相流模型）面板中打开 Implicit Body Force（隐式彻体力）选项。

用 VOF 模型计算自由表面运动时，应该在自由表面附近采用四边形网格（二维计算）或六面体网格（三维计算）。如果在全部流场上划分四边形网格或六面体网格有困难，也应该采用混合网格的形式，在自由表面附近区域采用这两种网格进行计算。

7.8.4 混合物模型

混合物模型中主要相和次要相的定义过程与 VOF 模型基本一样，区别是在定义次要相时可以定义颗粒的直径。

在 Phase Interaction（相间干扰）面板上，可以定义滑移速度和阻力函数。滑移速度在 Slip（滑移）标签下定义，可以选择的计算方法包括代数滑移模型（manninen-et-al，即 manninen 等人提出的模型）、无滑移模型和 user-defined（用户自定义）模型。阻力函数在 Drag（阻力）标签下定义，其中的阻力函数是欧拉模型中阻力函数集的一个子集，具体情况请参见下个段落的描述。

7.8.5 欧拉模型

欧拉模型中主要相的定义过程与混合物模型完全一致，非颗粒次要相（Non-Granular Secondary Phase）的定义与混合物模型的次要相定义也完全一致，区别在于对颗粒次要相（Granular Secondary Phase）的定义。

颗粒次要相的设置过程为：

1）在 Phase（物相）列表中选择 phase-2；

2）点击 Set...（设置）按钮打开 Secondary Phase（次要相）面板；

3）在 Name（名称）栏中设定物相名称；

4）在 Phase Material（物相材料）列表中选择物相材料；

5）定义材料性质；

6）打开 Granular（颗粒）选项；

7）如果要冻结颗粒相的速度场，则打开 Packed Bed（充填床）选项。打开这个选项后，在流场的其他区域也要将颗粒相的速度定义为 0。

8）在 Secondary Phase（次要相）面板中定义下列参数，即直径、颗粒粘度、体积粘度、摩擦粘度、内摩擦角、颗粒扩散系数、最大体积百分比等参数，这些参数均用于计算相间动量和能量交换过程计算。

9）点击 OK 按钮完成设置过程。

欧拉模型的相间干扰设置包括阻力函数设置、碰撞恢复系数设置、升力设置和虚拟质量力设置等内容。首先，欧拉模型允许在每对物相间设置阻力函数，具体过程如下：

1）在 Phase Interaction（相间干扰）面板中，点击 Drag（阻力）标签；

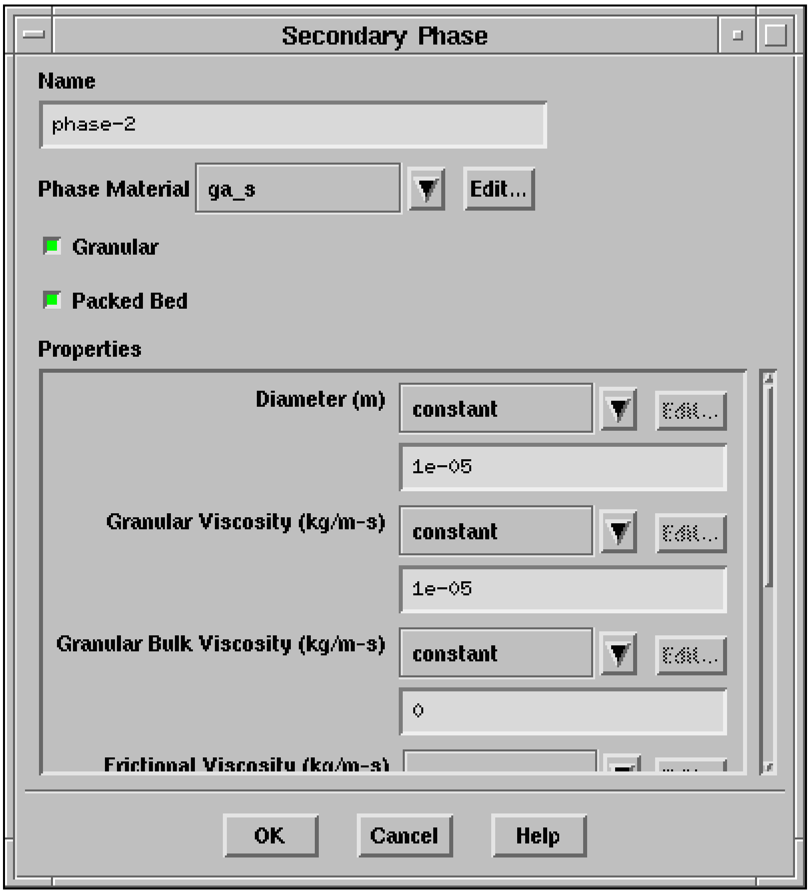


图 7-24 颗粒次要相的定义

2）针对每对物相，在下拉菜单中选择阻力函数。其中包括 schiller-naumann 模型、morsi-alexander 模型、symmetric（对称）模型等用于流体与流体之间阻力计算的模型，也包括 wen-yu 模型、gidaspow 模型、syamlal-obrien 模型等用于液体与固体之间阻力计算的模型，还包括 syamlal-obrien-symmetric 模型用于固体与固体之间的阻力计算。除此之外，还可以将阻力函数定义为 constant（常数），或者选择 user-defined（用户定义）由用户自己定义阻力函数。如果计算中不需要设定阻力，还可以选择 none（不计阻力）选项。

碰撞恢复系数在 Collisions（碰撞）标签中设定，缺省情况下这个系数被设定为 0.9，在多数情况下无需修改此设置。

升力系数在 Lift（升力）标签中设定，可以选择 none（无）、constant（常数）和 user-defined（用户定义）等方式定义升力。

虚拟质量力用于模拟颗粒加速过程中的惯性力。如果需要考虑虚拟质量力效应，就在 Phase Interaction（相间干扰）面板的左上角打开 Virtual Mass（虚拟质量）选项。

（4）在使用欧拉模型时，如果流动是湍流，则定义多相流的湍流模型，菜单操作为：

Define -> Models -> Viscous...

在 Viscous Model（粘性模型）面板中，可以选择三种形式的 $k-\varepsilon$ 模型进行湍流计算，

同时还可以针对多相流特点对湍流模型进行设置，其中包括 Mixture（混合物）选项用于计算混合物模型的湍流计算，Dispersed（稀疏）选项用于次要相非常稀释情况的多相流计算，还有 Per Phase（全部物相），用于湍流扩散作用在流场中占优势的多相流计算。

（5）如果流动中需要考虑彻体力作用，则打开重力选项，并定义重力加速度：

Define -> Operating Conditions...

FLUENT 中的"重力"，只要适当定义其数值和方向，就可以用来代表所有彻体力的合力。在 VOF 计算中，还应该同时在 Operating Conditions（操作条件）面板中打开 Specified Operating Density（指定的操作密度）选项，并将 Operating Density（操作密度）定义为比重最小的物相的密度。如果计算中设定了可压缩物相，则操作密度应该设为 0。同时建议在 Multiphase Model（多相流模型）面板中，打开 Body Force Formulation（彻体力计算方法）中的 Implicit Body Force（隐式彻体力）选项。

（6）在流场边界上定义边界条件。在使用 VOF 模型时，在边界上需要定义次要相的体积百分比，如果考虑壁面黏附作用，还要考虑壁面接触角：

Define -> Boundary Conditions...

多相流边界边界条件设置与单相流模型的设置过程略有区别，即除了要设定混合物的边界条件外，还要为混合物中的各种成分单独进行边界条件设置。

（7）设置求解参数：

Solve -> Controls -> Solution...

用 VOF 模型求解时间相关问题时，需要在 Solver（求解器）面板下面激活 Unsteady（非定常）选项，并选择合适的非定常算法。另外在缺省设置中，体积百分比在每个时间步中求解一次，而不是每次迭代计算一次。如果需要每次迭代，求解一次体积百分比，可以在 VOF Parameters（VOF 参数）下面打开 Solve VOF Every Iteration（每次迭代求解 VOF）选项。如果采用了滑移网格技术或动态网格技术，使用 Solve VOF Every Iteration（每次迭代求解 VOF）选项可以获得更精确的结果。

（8）在初始化流场后，可以为各物相设置初始分布。在非定常计算中，这个初始场就是初始流场；对于定常流计算，设置初始分布可以增加计算初始阶段的稳定性。可以在 Patch（补丁）面板中，用打补丁的方法为次要相设置初始体积百分比：

Solve -> Initialize -> Patch...

如果需要打补丁的区域已经被定义为独立的网格区域，可以直接将体积百分比的值定义在这个区域上。如果没有定义这样的区域，可以在网格中创建标志区域（register），然后再将体积百分比的值定义在上面。

（9）计算并检验计算结果。

在 VOF 计算中应该将参考压强点（reference pressure location）设在密度最小的地方，如果流场中存在气体区域，则应该将参考压强点设在气体区域中。压强计算应该采用 PRESTO!格式。体积百分比计算应该采用二阶格式或 QUICK 格式。在非定常计算中，推荐使用 PISO 格式，并且所有亚松弛因子可以设置为 1，而在使用四面体网格或三角形网格时，则建议将亚松弛因子设为 0.7 到 0.8 之间。如果计算中采用的是定常隐式 VOF 格式，则可以将所有亚松弛因子设在 0.2 到 0.5 之间。

在混合物模型计算中，可以先计算一个初始流场，然后再将多相流模型加入计算。方法是先在 Solution Controls（求解控制）面板中关闭 Volume Fraction（体积百分比）和 Slip Velocity（滑移速度）选项，在流场计算收敛后，再重新打开这些选项，并继续计算获得混合物模型的最终计算结果。

欧拉模型的计算策略类似于混合物模型的计算策略，区别在于初始流场的计算可以用欧拉模型中的主要相进行，也可以先用混合物模型进行。在获得初始流场后，再换用欧拉模型进行计算，直至获得最终的解。在需要考虑升力和虚拟质量力的问题中，也可以先不考虑升力和虚拟质量力的影响，在获得初始流场后，再将这些选项加入。

计算结束后，可供后处理的变量包括 VOF 模型的体积百分比（volume fraction）、混合物模型的颗粒直径和体积百分比、欧拉模型中的与颗粒动量、能量交换相关的一系列参数。

7.9　固化与熔化模型

FLUENT 采用“焓－多孔度（enthalpy-porosity）”技术模拟流体的固化和熔化过程。在流体的固化和熔化问题中，流场可以分成流体区域、固体区域和两者之间的糊状区域。“焓－多孔度”技术采用的计算策略是将流体在网格单元内占有的体积百分比定义为多孔度（porosity），并将流体和固体并存的糊状区域看作多孔介质区进行处理。在流体的固化过程中，多孔度从 1 降低到 0；反之，在熔化过程中，多孔度则从 0 升至 1。“焓－多孔度”技术通过在动量方程中添加汇项（即负的源项）模拟因固体材料存在而出现的压强降。

“焓－多孔度”技术可以模拟的问题包括纯金属或二元合金中的固化、熔化问题、连续铸造加工过程等。计算中可以计算固体材料与壁面之间因空气的存在而产生的热阻，固化、熔化过程中组元的输运等等。需要注意的是，在求解固化、熔化问题的过程中，只能采用分离算法，只能与 VOF 模型配合使用，不能计算可压缩流，不能单独设定固体材料和流体材料的性质，同时在模拟带反应的组元输运过程时，无法将反应区限制在流体区域，而是在全流场进行反应计算。

7.9.1 设置过程

固化、熔化问题的设置过程如下：

（1）启动 Solidification and Melting（固化与熔化）面板：

Define -> Models -> Solidification & Melting...

并打开 Solidification/Melting（固化、熔化）选项。FLUENT 将同时自动启动能量方程的计算，因此无需再手动启动能量方程计算。

（2）在 Parameters 下面定义 Mushy Zone Constant（糊状区域常数）。这个常数的取值范围一般在 10^4 到 10^7 之间，取值越大沉降曲线就越陡峭，固化过程的计算速度就越快，但是取值过大容易引起计算振荡，因此需要在计算中通过试算获得最佳数值。

（3）如果在计算中需要计算固体材料的“拉出速度（pull velocity）”，则要打开 Include Pull Velocities（包含拉出速度）选项。

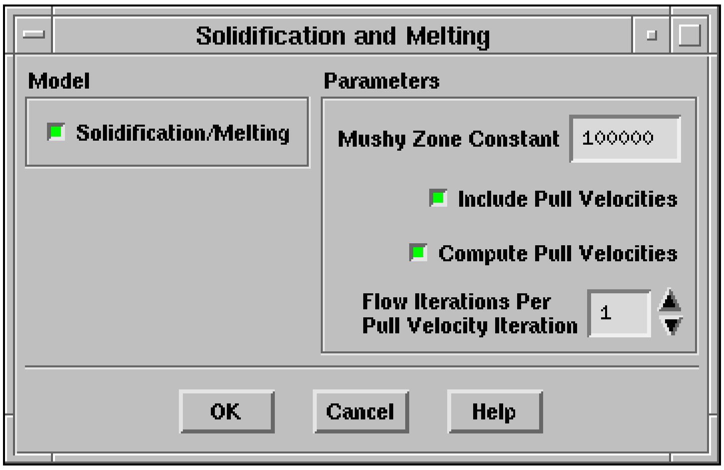


图 7-25 Solidification and Melting（固化与熔化）面板

（4）在计算拉出速度的同时，如果希望用速度边界条件推算拉出速度，则打开 Compute Pull Velocities（计算拉出速度）选项，并定义 Flow Iterations Per Pull Velocity Iteration（拉出速度迭代一次对应的流场迭代次数）。在缺省情况下，流场每迭代一次计算一次拉出速度，即上述选项取值为 1。如果将这个数值增加为 2，意味着流场每迭代二次计算一次拉出速度。在固化过程即将结束时，可以增加这个参数值，以便提高计算速度。

（5）在 Materials（材料）面板上，定义 Melting Heat（熔化热）、Solidus Temperature（固相点温度）和 Liquidus Temperature（液相点温度）。如果计算中涉及组元输运过程，则必须同时定义溶剂的融解温度（Melting Temperature），同时需要定义熔化物的液相线相对于浓度的斜率（Slope of Liquidus Line）、分配系数（Partition Coefficient）和固体中的扩散速率（Diffusion in Solid）等参数。

（6）设置边界条件。除了常规的边界条件设置，对于固化和熔化问题还有一些特殊设置，其中包括：

- 在计算壁面接触热阻时设置接触热阻（Contact Resistance）。这个参数在 Wall（壁面）面板中的 Thermal Conditions（热力学条件）下给定。
- 如果需要定义壁面上表面张力对温度的梯度，则在 Shear Condition（剪切条件）下选择 Marangoni Stress（Marangoni 应力）选项。
- 如果计算拉出速度，则在边界条件中的速度边界条件将被用于拉出速度的计算。

7.9.2 连续铸造的求解流程

FLUENT 中有三种方法计算连续铸造过程：

（1）在流场初始化结束后，用打补丁（patch）方法将拉出速度定义为常数或函数。

（2）用速度边界条件计算拉出速度。只要打开 Compute Pull Velocities（计算拉出速度）选项，FLUENT 就可以开始对拉出速度的计算。因为这种方法消耗系统资源比较大，所以一般只在拉出速度强烈地依赖于液固交界面位置的时候才使用。

（3）仅在一个迭代步中计算拉出速度，然后把计算结果用于余下的计算中。方法是在开始计算时先打开 Compute Pull Velocities（计算拉出速度）选项，迭代一步，然后关闭这个选项并继续进行计算。

固化和熔化过程也是高度非线性的过程，如同其他非线性问题一样，处理这类问题的一个基本策略就是从简到繁——先避开一些复杂模型算出初始流场，再添加各种模型进行精细计算。这样逐步增加问题的复杂度，才能在稳定性、精度和计算速度之间找到平衡。

7.10　气动噪声模型

气动噪音的生成和传播可以通过求解可压 NS 方程的方式进行数值模拟。然而与流场流动的能量相比，声波的能量要小几个数量级，客观上要求气动噪音计算所采用的格式应有很高的精度，同时从音源到声音测试点划分的网格也要足够精细，因此进行直接模拟对系统资源的要求很高，而且计算时间也很长。为了弥补直接模拟的这个缺点，可以采用 Lighthill 的声学近似模型，即将声音的产生与传播过程分别进行计算，从而达到加快计算速度的目的。

FLUENT 中用 Ffowcs Williams 和 Hawkings 提出的 FW-H 方程模拟声音的产生与传播，这个方程中采用了 Lighthill 的声学近似模型。FLUENT 采用在时间域上积分的办法，在接收声音的位置上，用两个面积分直接计算声音信号的历史。这些积分可以表达声音模型中单极子、偶极子和四极子等基本解的分布。积分中需要用到的的流场变量包括压强、速度分量和音源曲面的密度等等，这些变量的解在时间方向上必须满足一定的精度要求。满足时间精度要求的解可以通过求解非定常雷诺平均方程（URANS）获得，也可以通过大涡模拟（LES）或分离涡模拟（DES）获得。音源表面既可以是固体壁面，也可以是流场内部的一个曲面。噪音的频率范围取决于流场特征、湍流模型和流场计算中的时间尺度。

FLUENT 允许用户选择多重音源曲面、多个声音接收点。可以将音源数据保存起来以备将来使用，也可以在非定常计算进行过程中还可以随时进行声学计算。计算中得到的声音压强信号可以用快速 Fourier 变换进行处理，也可以在后处理中计算声音压强的总体水平和功率谱。

在气动噪音的计算中只能采用分离求解器，并可以计算远场噪音，但是不能计算封闭空间中的噪音，同时计算中只能使用静止的音源表面，即 FLUENT 中的声学模型可以模拟静止汽车外部的噪音，而无法模拟汽车内部的噪音。

气动噪声的计算大体上可以分为两大步：首先通过流场计算，求出满足时间精度要求的各相关变量（压强、速度和密度）在音源曲面上的变化过程；然后利用求出的音源数据计算声音接收点处的声音压强信号。

在用 FW-H 方程通过积分计算声音压强时，FLUENT 用“前向时间投射法（forward-time projection）”将声音从发射到接收之间的时间延迟考虑进去，从而可以在非定常流场计算的同时计算声音压强，而不必再保存音源的数据。

噪音计算的设置过程如下：

（1）通过非定常计算获得“统计学意义上的稳态解”，即流场变量的统计量不再随时间变化。

（2）启用声学模型并设置相关的模型参数：

Define -> Models -> Acoustics -> Models & Parameters...

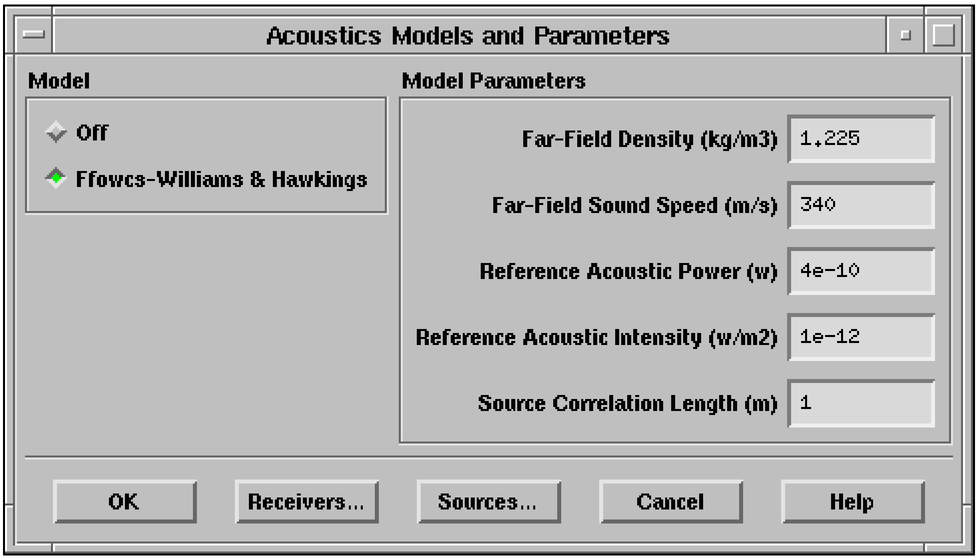


图 7-26　Acoustics Models and Parameters（声学模型与参数）面板

在面板中点击选择 Ffowcs-Williams & Hawkings（即 FW-H 算法），则面板自动展开出现相关参数输入栏。这些参数包括 Far-Field Density（远场密度）、Far-Field Sound Speed（远场声速）、Reference Acoustic Power（参考声功率）、Reference Acoustic Intensity（参考声强度）和 Source Correlation Length（声源修正长度）。其中声源修正长度是在使用二维流场计算结果计算声音信号时，FW-H 积分中用来代替深度方向长度的值。

（3）定义音源曲面，并打开与获取、保存音源数据有关的选项：

Define -> Models -> Acoustics -> Sources...

在 Acoustic Source Specification（声源定义）面板上，可以在 Source Zones（声源区域）中选择声源曲面，并在 Type（类型）中定义声源区域的类型。

在 FLUENT 中允许定义多个声源曲面，但是声源曲面不允许互相阻挡各自的声音传播路线，更不允许一个曲面将另一个曲面包围。如果声源曲面是位于流场内部，则曲面两侧的网格区域必须采用不同的编号，同时需要指定代表声源的基本解（四极子）位于哪个网格区域的边界上，指定四极子就可以指定声音的传播方向。

如前面所述，FLUENT 可以在计算声音压强信号时采用“前向时间投射”技术，从而

避免在计算过程中执行保存声源数据的操作。如果需要在计算中采用这项技术，可以在Acoustic Source Specification（声源定义）面板上打开 Extract Acoustic Signals Simultaneously（同时计算声音信号）选项。

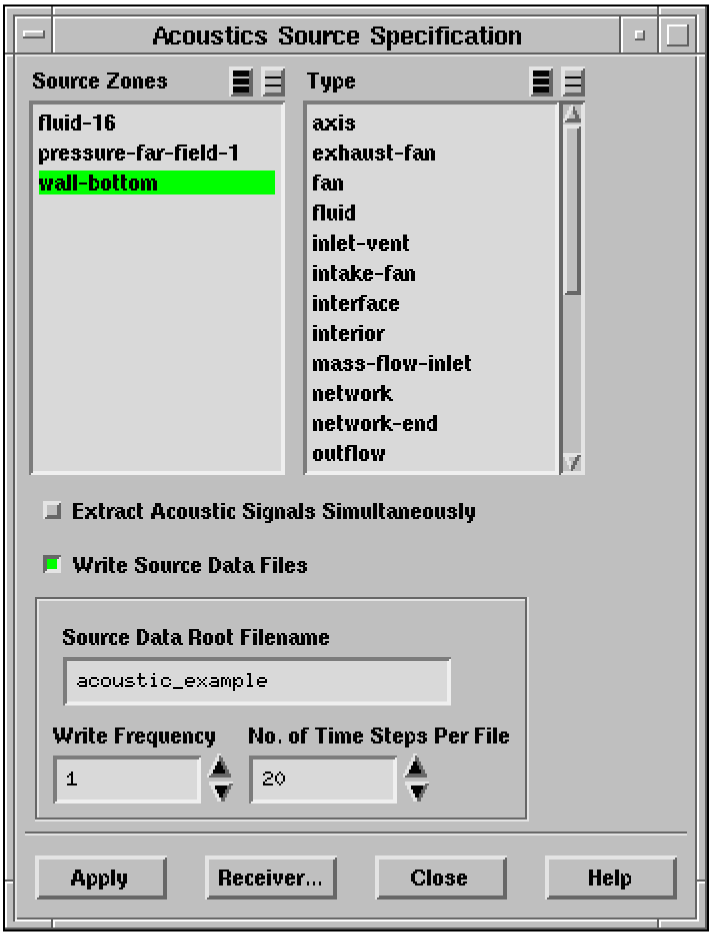


图 7-27 Acoustic Source Specification（声源定义）面板

保存声源数据虽然会占用大量硬盘，但是将计算所得到的数据存盘后，可以在以后的工作中随时调用这些数据用于声音信号的计算，所以很多时候保存声源数据也是必要的。如果想保存这些数据，可以打开 Write Source Data Files（写声源数据文件）选项。

（4）指定声音接收点的位置：

Define -> Models -> Acoustics -> Receivers...

在 Acoustics Receivers Specification（声音接收点定义）面板上对声音接收点进行定义。首先在 No. of Receivers（接收点数量）内确定接收点的数量，然后在下面栏目中指定接收点的坐标和保存文件的名称。

（5）让非定常计算运行足够长的时间，并保存音源数据。

（6）计算并保存声音压强信号：

Report -> Acoustics -> Read & Compute Sound...

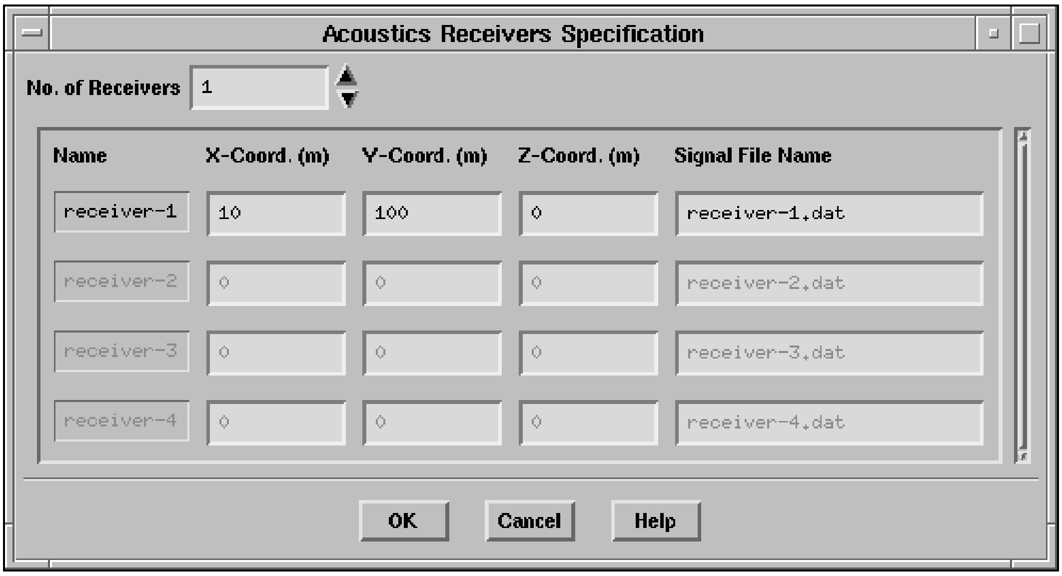


图 7-28　Acoustics Receivers Specification（声音接收点定义）面板

在 Acoustic Signal Computation（声音信号计算）面板中计算声音数据，方法如下：

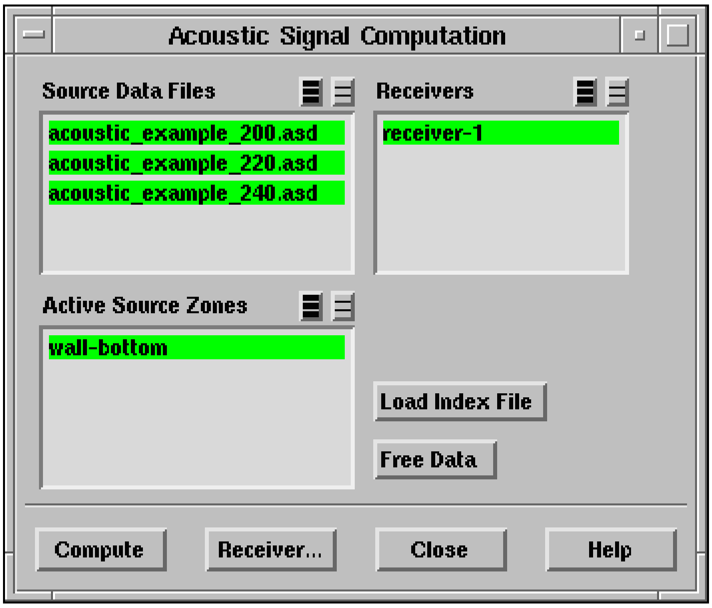


图 7-29 Acoustic Signal Computation（声音信号计算）面板

1）点击 Load Index File（加载索引文件），并在 Select File（选择文件）对话框中选择索引文件——索引文件名与用户在 Acoustic Source Specification（声源定义）面板中填写在

Source Data Root Filename（声源数据根文件名）中的一样，文件名的后缀为.index，比如前面填写的文件名是 acoustic，则其索引文件就是 acoustic.index。在索引文件的使用过程中可以随时点击 Free Data（释放数据）按钮，将与索引文件相关联的数据从内存中释放。



图 7-30 Write Sound Signals（写声音信号）面板

2）在 Source Data File（声源数据文件）列表中，选择用于声音信号计算的声源文件，声源文件名的后缀是.asd。



图 7-31 Fourier Transform（Spectral Analysis）面板

3）在 Active Source Zones（激活声源区域）列表中，选择需要计算声音信号的区域。

4）在 Receivers（接收点）列表中，选择需要计算声音信号的接收点。

5）点击 Compute（计算）按钮开始计算并保存声音压强数据文件，结果保存在接收点数据文件中，这个数据文件就是在 Acoustic Signal Computation（声音信号计算）面板中指定的文件 receiver-1.dat。

6）如果计算中选择采用了“前向时间投射”技术，则仅需保存声音压强数据文件。这项操作是在 Write Sound Signals（写声音信号）面板中完成操作的。这个面板的打开方法为：

Report -> Acoustic -> Write Sound Pressure...

点击 Write...（写）按钮，则每个接收点上的数据都被保存在各自的文件中。

（7）对声音压强信号进行后处理。Fourier Transform（Spectral Analysis）面板的启动方法为：

Plot -> FFT...

在 Fourier Transform（Spectral Analysis）——Fourier 变换（谱分析）面板上，点击 Load Input File...（加载输入文件）并选择适当的.dat 文件，然后点击 Plot FFT（FFT 操作）按钮，既开始 FFT 计算和曲线绘制。

7.11 热交换模型

流体间热交换的形式有热传导、对流换热和辐射换热三种基本形式，其中热传导和对流换热的计算相对简单，而辐射换热要复杂一些。除了上述三种换热形式，FLUENT 中还可以计算周期性几何条件下的换热问题和浮力驱动流动中的换热问题。同辐射换热一样，这两种形式的换热计算也比较复杂。下面分别介绍上述几种换热问题的计算方法。

7.11.1 对流换热和热传导计算

计算换热问题同计算其他问题一样，都需要进行激活相关的物理模型、设定边界条件、确定材料属性等相关操作。下面介绍热传导问题的设置过程：

（1）首先在 Energy 面板中激活 Energy Equation 选项。

（2）加入粘性加热项。在流体中剪切力很大的流动问题时，比如在计算高速、可压缩流动时，应该能量方程中加入粘性加热项。因为耦合求解器中粘性加热项是自动加热的，而分离算法则没有自动加入粘性加热项，所以在使用分离求解器时，需要在 Viscous Model（粘性模型）面板中手动打开 Visous Heating（粘性加热）选项。

（3）在流场的入口、出口和壁面上设置热力学边界条件。在入口和出口条件中可以设置温度，在壁面上可以设置热流通量、温度、对流换热、外部辐射以及外部辐射与对流的混合边界条件。

（4）定义材料属性。在定义材料属性时，热容和导热系数是必须定义的。在定义过程中，也可以根据需要将各种材料属性定义为温度的函数。

在 Solution Limits（解的极限）面板中，可以设定流场中温度的最大值和最小值，这两个量的缺省值分别是 5000K 和 1K。

如果计算的流场是两个完全分开的区域，中间被固体壁面隔开的话，则在设置过程中不能使用 outflow（出流）边界条件。在两个区域上可以采用完全不同的流体介质，但是如果涉及组元计算，则整个区域只能被设置为同一种混合物材料作为流体介质。

在采用分离求解器计算热传导时，能量方程和温度的亚松弛因子在缺省情况下均被设为 1，如果计算不收敛，可以降低亚松弛因子的值。另外还可以考虑在 Species Model（组元模型）面板中，取消对 Diffusion Energy Source（扩散能量源项）的选择，这样可以去掉扩散对计算的影响，增加计算的稳定性。不过在耦合求解器中，这一项是自动耦合在算法中，因此是无法取消的。

如同计算其他物理模型一样，为了加速计算或增加计算的稳定性，可以先将计算中涉及到的物理模型去掉，单独计算流场。比如可以将能量方程关闭，在流场计算收敛后，再打开能量方程开始下一轮计算，直至计算收敛。

7.11.2 辐射换热

FLUENT 中可以用 5 种模型计算辐射换热问题。这 5 种模型是离散换热辐射模型（DTRM）、P-1 辐射模型、Rosseland 辐射模型、表面辐射（S2S）模型和离散坐标（DO）辐射模型。可以计算的问题包括火焰辐射，表面辐射加热或冷却，辐射、对流和热传导的耦合换热问题，空调、通风设备中通过窗口的辐射换热，汽车车厢内的热交换分析，玻璃加工、玻璃纤维拉丝和陶瓷加工过程中的辐射换热等等。辐射换热是高温换热的主要机制，因此在计算高温换热问题时应该采用辐射换热模型。

选择辐射模型的标准

可以根据下列标准选择辐射模型：

（1）光学厚度：可以用光学厚度（optical thickness）αL 作为选择辐射模型的一个指标，L 为流场的特征长度尺度。在 αL 远远大于 1 时，最好选用 P-1 和 Rosseland 辐射模型进行计算。P-1 模型一般用于光学厚度大于 1 的情况。在光学厚度大于 3 时，则应该选用计算量更小而且效率更高的 Rosseland 模型。DTRM 和 DO 模型对于任何光学厚度都适用，但是它们的计算量较大，因此计算中应尽可能选择 P-1 或 Rosseland 辐射模型。对于光学厚度比较小的问题（αL 小于 1），则只能用 DTRM 和 DO 模型进行计算。

（2）散射与发射率：P-1、Rosseland 和 DO 模型均可以计算散射问题，而 DTRM 模型则忽略了散射的影响。Rosseland 模型在壁面上使用了温度滑移边界条件，因此它对壁面的发射率不敏感。

（3）颗粒效应：只能用 P-1 和 DO 模型计算气体与颗粒之间的辐射换热。

（4）半透明介质与镜面边界：只有 DO 模型可以计算镜面反射和半透明介质内的辐射。

（5）非灰体辐射：只有 DO 模型可以通过灰带模型计算非灰体辐射。

（6）局部热源：在带有局部热源的问题中，DO 模型是这种问题最合适的计算方法。如果采用足够多的射线，也可以用 DTRM 模型进行计算。用 P-1 模型计算的辐射热通量则可能高于正常值。

（7）封闭腔体内没有辐射介质时的辐射传热：表面辐射（S2S）模型适用于这种问题的计算。

（8）外部辐射：如果辐射源在计算域外面的话，可以在边界条件中定义外部辐射边界条件。如果计算域内没有辐射源，则无需在计算中设定任何辐射模型，只要设置好外部辐射边界条件就可以进行计算了。

各种辐射模型的优点和局限

（1）DTRM 模型：DTRM 模型的优点是比较简单，通过增加射线数量就可以提高计算精度，同时还可以用于很宽的光学厚度范围。其局限包括：

- DTRM 模型假设所有表面都是漫射表面，即所有入射的辐射射线没有固定的反射角，而是均匀地反射到各个方向。
- 计算中没有考虑辐射的散射效应。
- 计算中假定辐射是灰体辐射。
- 如果采用大量射线进行计算的话，会给 CPU 增加很大的负担。

（2）P-1 模型：相对于 DTRM 模型，P-1 模型有一定的优点。对于 P-1 模型，辐射换热方程（RTE）是一个容易求解的扩散方程，同时模型中包含了散射效应。在燃烧等光学厚度很大的计算问题中，P-1 的计算效果都比较好。P-1 模型还可以在采用曲线坐标系的情况下计算复杂几何形状的问题。P-1 模型的局限如下：

- P-1 模型也假设所有表面都是漫射表面。
- P-1 模型计算中采用灰体假设。
- 如果光学厚度比较小，同时几何形状又比较复杂的话，则计算精度会受到影响。
- 在计算局部热源问题时，P-1 模型计算的辐射热流通量容易出现偏高的现象。

（3）Rosseland 模型。

同 P-1 模型相比，Rossland 模型的优点是不用象 P-1 模型那样计算额外的输运方程，因此 Rosseland 模型计算速度更快，需要的内存更少。Rosseland 模型的缺点是仅能用于光学厚度大于 3 的问题，同时计算中只能采用分离求解器进行计算。

（4）DO 模型。

DO 模型是适用范围最大的模型——它可以计算所有光学厚度的辐射问题，并且计算范围涵盖了从表面辐射、半透明介质辐射到燃烧问题中出现的介入辐射在内的各种辐射问题。DO 模型采用灰带模型进行计算，因此既可以计算灰体辐射，也可以计算非灰体辐射。如果网格划分不过分精细的话，计算中所占用的系统资源也不大，因此成为辐射计算中被经常使用的一个模型。

（5）表面辐射（S2S）模型。

S2S 模型适用于计算没有介入辐射介质的封闭空间内的辐射换热计算，比如太阳能集热器、辐射式加热器和汽车机箱内的冷却过程等。同 DTRM 和 DO 模型相比，虽然视角因数（view factor）的计算需要占用较多的 CPU 时间，S2S 模型在每个迭代步中的计算速度都很快。S2S 模型的局限如下：

- S2S 模型假定所有表面都是漫射表面。
- S2S 模型采用灰体辐射模型进行计算。
- 内存等系统资源的需求随辐射表面的增加而激增。计算中可以将辐射表面组成集群的方式减少内存资源的占用。
- S2S 模型不能计算介入辐射问题。
- S2S 模型不能用于带有周期性边界条件或对称性边界条件的计算。
- S2S 模型不能用于二维轴对称问题的计算。
- S2S 模型不能用于多重封闭区域的辐射计算，只能用于单一封闭几何形状的计算。

辐射模型的设置

（1）选择辐射模型。

Radiation Model（辐射模型）面板的启动方法为：

Define -> Models -> Radiation...

在 Model（模型）下面从五种模型中选择一种作为辐射计算模型，同时也可以通过选择 Off（关闭）从计算中去掉辐射模型。在选择 DTRM、DO 或 S2S 模型时，Radiation Model（辐射模型）面板会展开，并出现与这些模型相关的参数设置选项。需要注意的是，在选用任何一个辐射模型后，系统都会自动将能量方程加入计算，因此不必再去打开能量方程的选项。

（2）设置与 DTRM、S2S 和 DO 模型相关的参数。

如果选择了 DTRM 模型，则面板会展开并出现 Flow Iterations Per Radiation Iteration（每次辐射迭代之间的流场迭代次数）、Number of DTRM Sweeps（DTRM 扫描次数）和 Tolerance（容差）三个计算控制参数输入栏。保持缺省设置或根据计算需要填入相应的数据后，点击 OK 按钮进入 Ray Tracing（射线跟踪）面板，然后定义相应的射线跟踪参数。

用菜单打开 Ray Tracing（射线跟踪）面板的方法如下：

Define -> Ray Tracing...

在面板中填入 Cells Per Volume Cluster（每个体积集群中的网格单元数量）和 Faces Per Surface Cluster（每个表面集群中的网格面数量），然后再在 Theta Divisions（θ分区）和 Phi Divisions（ϕ分区）中设定射线数量。参数设定完毕后，点击 OK 按钮创建、保存 DTRM 射线文件。然后，Display Clusters（显示集群）按钮查看射线的设定，如果设置有问题，可以重新进行设置。在对网格进行过编辑后，需要重新创建射线文件。在计算中重新读入射线文件的方法为：

File -> Read -> Rays...

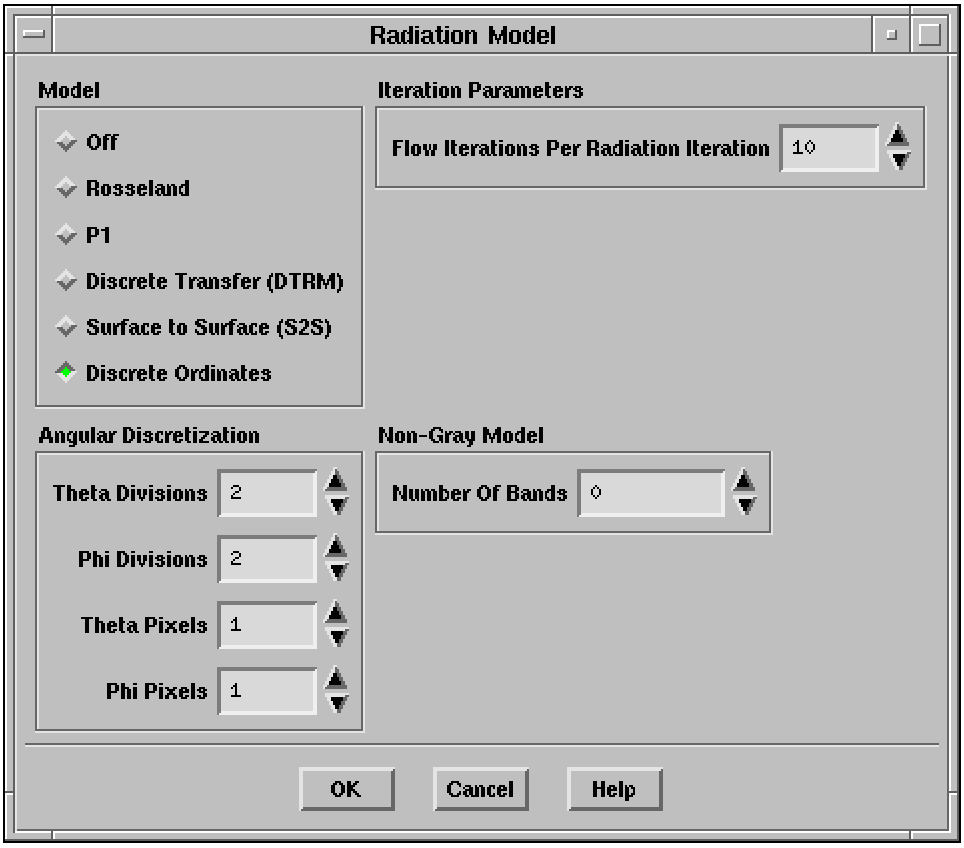


图 7-32 Radiation Model（辐射模型）面板

在射线的设置过程中如果点击 Cancel 按钮则取消所有与 DTRM 相关的设置。

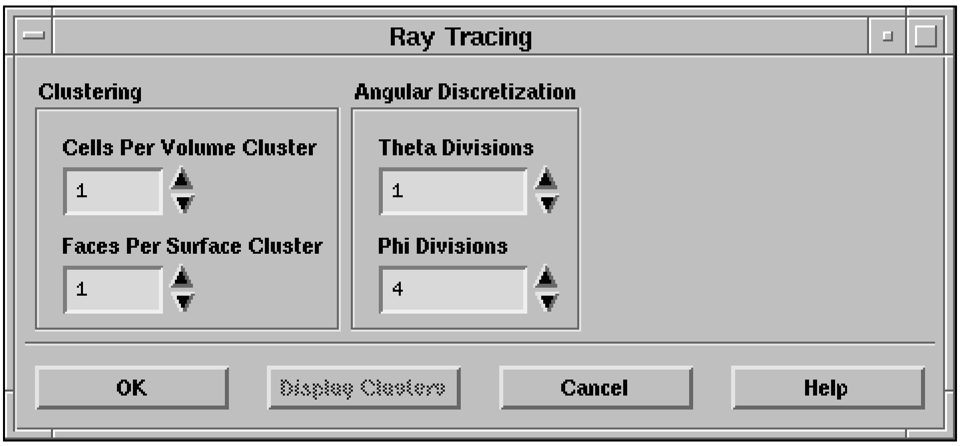


图 7-33 Ray Tracing（射线跟踪）面板

如果选择了 S2S 模型，则 Radiation Model（辐射模型）面板展开如下图。类似于 DTRM

模型中的情况，首先在这个面板中的 Iteration Parameters（迭代参数）下面设置计算控制参数。然后在 View Factors（视角因数）组内的 Parameters（参数）下面点击 Set...（设置）按钮进入 View Factors and Clusters Parameters（视角因数和集群参数）面板，设置视角参数和集群参数。参数设置完成后，点击 OK 按钮退出到 Radiation Model（辐射模型）面板，点击 Comput/Write（计算或写文件）按钮计算视角因数，并将结果输出到文件中。

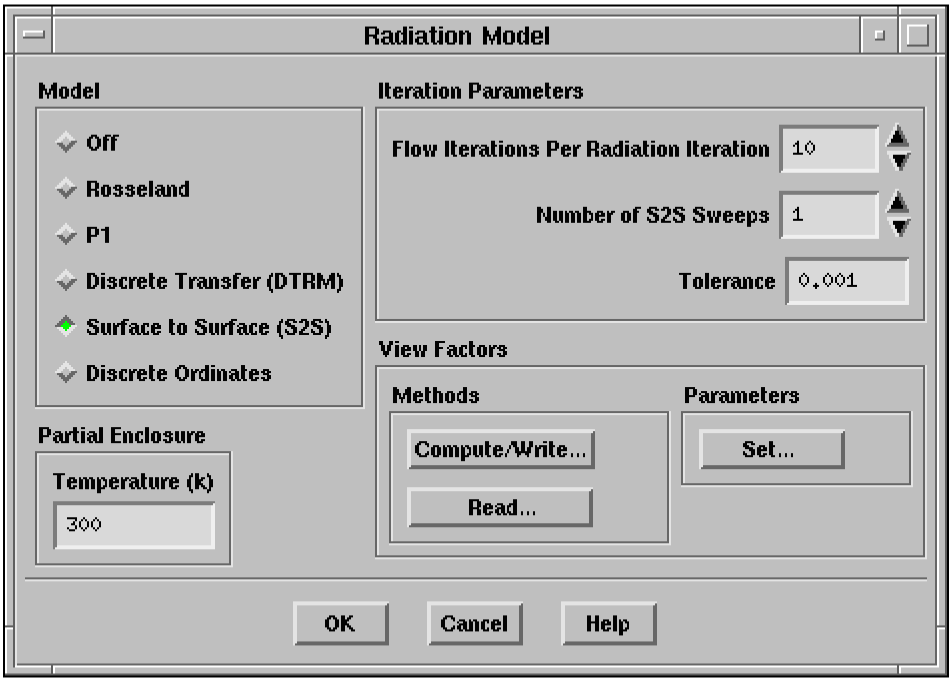


图 7-34 Radiation Model（辐射模型）面板

如果选择了 DO 模型，Radiation Model（辐射模型）面板自动展开并出现与迭代计算、角向离散和非灰度模型相关的参数设置栏，其设置与上面两个模型中相应的设置类似，用户可以根据计算需要进行设置。

（3）定义材料性质。

如果计算中选用了 P-1、DO 或 Rosseland 辐射模型，则应该 Materials（材料）面板中定义流体的吸收系数和散射系数。如果用 DO 模型计算半透明介质，则需要定义半透明流体或固体的折射系数。对于 DTRM 模型则只需定义吸收系数。

如果气相组元中包含燃烧产物，则气体中的吸收和散射会大大增强。如果流体中含有弥散相粒子或液滴，则需要相应地加大散射系数。FLUENT 中可以用 WSGGM（Weighted Sum of Grey-Gas Model，灰度气体加权求和模型）计算混合物的吸收系数。

如果计算中采用的是非灰度 DO 模型，则可以为灰带模型上的每个条带设置一个常值吸收系数，而不能为每个条带设置一个随组分变化的吸收系数。如果用 WSGGM 法计算一个变化的吸收系数，则这个系数同时适用于所有条带，而不是特定的条带。在每个条带上同时可以设定常值的折射系数。

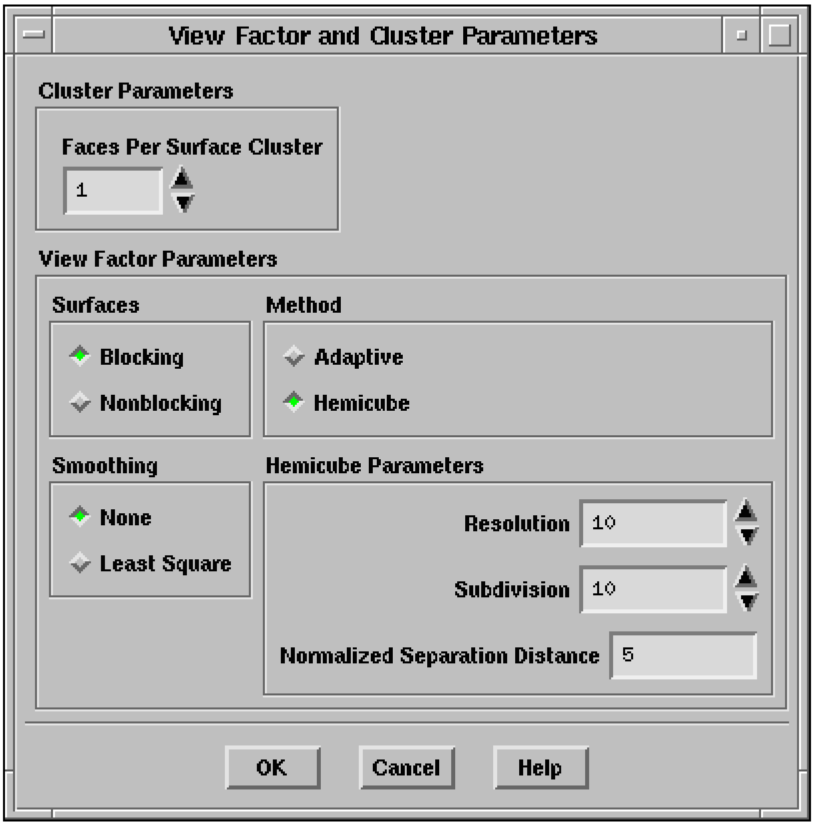


图 7-35 View Factor and Cluster Parameters（视角因数和集群参数）面板

（4）定义边界条件。如果模型中包含半透明介质，同时设置半透明介质。

在采用辐射模型进行计算时，需要在边界条件中增加一些设置。在入口和出口边界上需要增加 Internal Emissivity（内部发射率）和 Black Body Temperature（黑体温度）。发射率的缺省值是 1，在多数计算中不必改动这个值。黑体温度通常指入口和出口外部的温度，在外界温度远远大于内部流场温度时，可以将辐射换热的温度与对流计算的温度分别进行设置。因为高温区辐射总是强于低温辐射，所以在入口或出口外部温度远低于内部流场温度时需要小心设置黑体温度——如果没有把握，可以将黑体温度设为接近内部流场温度的值，而在外部温度远高于内部流场时则可以直接将黑体温度设为外部温度。需要注意的是，在采用 Rosseland 模型时无法设置这两个参数。

DTRM、P-1、S2S 和 Rosseland 模型均假设壁面为灰体和漫射表面。在 Wall（壁面）面板中可以设置的辐射边界条件就是发射率。Rosseland 模型的发射率是 1。DTRM、P-1 和 S2S 模型的发射率可以修改，其缺省值也是 1。另外在 S2S 模型中可以设置部分封闭的壁面边界条件，设置方法就是在 Wall（壁面）面板的 Thermal（热力学）标签下，取消对 Pariticipates in S2S Radiation（参与 S2S 辐射模型计算）选项的选择。不参与辐射模型计算的壁面被作为黑体进行计算。

在使用 DO 模型时，可以定义不透明（漫射或镜面反射）和半透明墙壁。壁面类型在

Wall（壁面）面板的 Radiation（辐射）标签下的 BC Type（边界条件类型）列表中进行选择。在设置了 opaque（不透明）壁面类型时，可以设定壁面的 Diffuse Fraction（漫射率），即有多少入射光被漫射掉了。漫射率的缺省值是 1，如果设置了小于 1 的漫射率，则意味着余下的入射光被“规则地反射”出去了。需要提醒的是，DO 模型中同样需要设置发射率。

在为 DO 模型选择了 semi-transparent（半透明）类型后，还需要进一步设置 Irradiation（放射通量）、Beam Width（光束宽度）和 Beam Direction（光束方向）。如果 DO 模型采用了非灰度模型，则放射通量对所有频段都是常数。半透明模型中也要设置漫射壁面的漫射率。Heat Flux（热流通量）条件也在 Thermal（热力学）标签下进行设置。

DO 模型只在流体区域中使用，而不包括固体区域。如果想计算半透明介质，让固体壁面参与辐射计算，则在 Solid（固体）面板中打开下部的 Participates In Radiation（参与辐射计算）选项就可以了。

对于双侧壁面来说，DO 模型可以分别将壁面的两侧设置为 diffuse（漫射）或 specular（镜面反射）。在 Wall（壁面）面板中可以指定 Wall Thickness（壁面厚度）和 Material Name（材料名称），同时指定壁面材料的折射率和吸收系数。

（5）在选用 DTRM、DO、S2S 或 P-1 模型时，需要设置相应的求解参数。

在采用 DTRM 模型时，可以通过改变 Number of DTRM Sweeps（DTRM 扫描次数）控制每次迭代中辐射计算的最大扫描次数。在缺省情况下，这个参数的值为 1，即每次迭代计算一次辐射强度。如果增加这个参数的值，则每次迭代中将多次计算辐射强度，直到计算满足了 Tolerance（容差）要求或计算次数达到设定值。Tolerance（容差）的缺省值是 0.001，在辐射强度增量小于这个参数值时，系统认为计算已经收敛。

Flow Iterations Per Radiation Iteration（每次辐射计算间隔的流场迭代次数）的缺省值是 10，意思是流场每迭代 10 次进行一次辐射计算。增加这个参数值可以加速计算，但同时使收敛过程放慢。

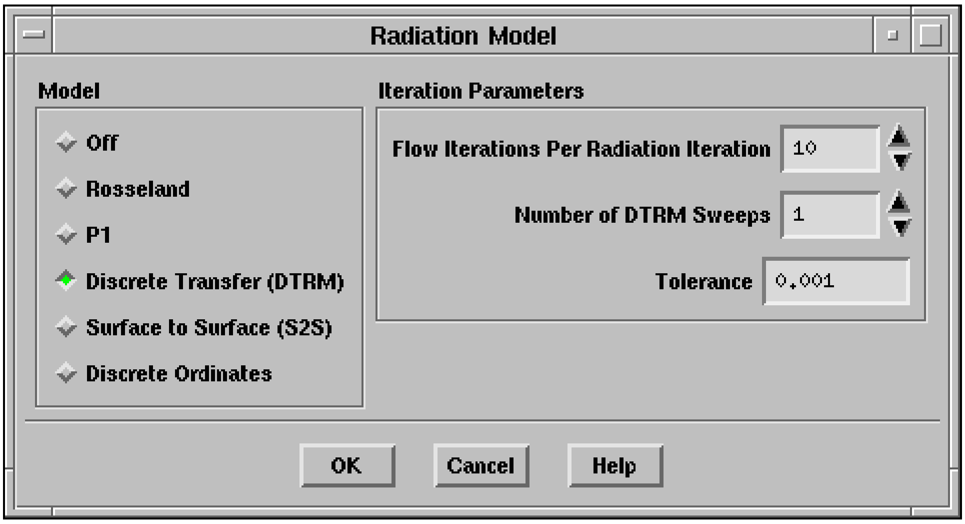


图 7-36 设置 DTRM 模型的求解参数

与 DTRM 模型类似，S2S 模型计算中也需要设定 Flow Iterations Per Radiation Iteration（每次辐射计算间隔的流场迭代次数）。在使用分离求解器计算初始流场时，应该将能量方程关闭，同时将这个参数从 10 降到 1 或 2。S2S 模型中还需要设定 Number of S2S Sweeps（S2S 扫描次数）和 Tolerance（容差），其含义与 DTRM 模型中的类似，这里不再赘述。

选用 DO 模型时，仅需设定 Flow Iterations Per Radiation Iteration（每次辐射计算间隔的流场迭代次数）即可。

选用 P-1 模型时，求解过程是通过收敛判据和亚松弛因子进行控制的，同时需要注意光学厚度。P-1 模型的缺省收敛判据是 10^{-6}，与能量方程的判据一样。为了使 P-1 模型能以最优化方式收敛，光学厚度 $(a+\sigma_s)L$ 必须介于 0.01 和 10 之间，最好不要超过 5。与较小的光学厚度对应的是 1 厘米左右的小开口，但是在这类问题中可以提高吸收系数的值以便使 $(a+\sigma_s)L$ 的值等于 0.01。这里“随意”提高吸收系数的值实际上不会影响问题的物理本质，因为对于光学厚度等于 0.01 和小于 0.01 的介质，其透明度的差值是可以忽略不计的。

（6）求解带辐射模型的流场。

在计算过程中，P-1 和 DO 模型的计算过程可以通过残差进行监视，而 DTRM 和 S2S 模型因为仅影响能量方程，所以其变化可以通过能量方程的残差反映出来的。除此之外，DTRM 和 S2S 模型在计算过程中以文字形式报告最新计算结果与前一步结果之间的差异，也可以利用这个数据判断计算的稳定性。

7.11.3 周期性热交换

本节讨论周期性流动问题中的热交换问题，即在流动方向上流场几何形状呈周期性变化流动问题中的热交换问题。流向周期性流动最常见的一个例子是沿流向排列的圆管阵列绕流问题——在流动方向上，圆管按一定的距离排列，同时压强降低一个常数值。根据流向周期性流动的特点，可以将每个圆管前后一定距离（即“周期长度”）内的流动区域切割出来，形成一个流场单元。通过分析这个单元上的流动就可以获得整个流场的流动情况，从而简化计算。

在 Boundary Conditions（边界条件）面板中的 Thermal（热力学）标签下，将壁面温度或热流通量设置为常数，就可以构成周期性流动的热力学条件。流向周期性流动问题必须用分离求解器进行计算，同时边界条件设置中不能采用温度和热流通量混合设置的形式，即要么将壁面条件全部设置为指定温度，要么全部设置为指定热流通量。设置为指定温度时，必须将所有壁面温度设置为同一个常数，而不能设置为温度分布形式；设置为热流通量时，则可以采用热流通量的分布形式。温度与热流通量混合设置的唯一可能情况是将温度设置为常数，同时热流通量设置为 0，这种设置其实与常值温度边界条件是等价的。在采用常值温度壁面条件时，计算中不能包含粘性加热效应或任何体积形态的热源。另外，流向周期性问题中，流场单元的边界必须是流体边界，即固体区域不能跨越流场单元的边

界。

流向周期性问题中，流体的热力学性质和输运性质，包括黏性、密度、导热系数等等，不许定义为温度函数，因此在周期性流动计算中就不能包含化学反应流动计算。流体的热力学性质可以周期性变化，因此可以模拟周期性变化的湍流流场。

流向周期性流动计算可以分为速度场计算和温度场计算两部分。计算过程中，首先冻结温度场，计算速度场；然后再冻结速度场，计算温度场。具体步骤大致如下：

（1）用平移周期性边界条件设置网格。

（2）输入常数的热力学性质或分子输运性质。

（3）在周期性边界上设定周期性压强梯度或净质量流量。

（4）计算周期性流场，求解动量、连续和湍流方程。

（5）用定义热流通量或常值温度的方式，定义壁面上的热力学边界条件。

（6）定义入口处气流温度。

（7）求解能量方程，获得周期性流场的温度场。

在 FLUENT 软件中的具体操作如下：

（1）在 Energy（能量）面板中激活能量方程计算。

（2）在壁面边界上，用定义壁面温度或热流通量的方式定义热力学边界条件。

（3）计算中可以将固体区域定义为热导体。如果用定义壁面温度的方式进行定义，则固体区域的边界上必须保持固定的温度，同时固体区域内不能有热量生成；如果用定义热流通量的方式进行定义，则可以在固体区域内定义热量生成。

（4）在 Materials（材料）面板中，将材料性质定义为常数，而不是温度的函数。

（5）在 Periodicity Conditions（周期性条件）面板中，定义 Upstream Bulk Temperature（上游气体温度）。Periodicity Conditions（周期性条件）面板启动方式为：

Define -> Periodic Conditions...

需要注意的是，在用指定温度的方式定义边界条件时，不要让上游气体温度与壁面温度相等，因为这样会得到全流场温度为常数的平凡解。

在所有条件设置完毕后，就可以开始进行计算了。如前所述，计算中经常采取的一个策略是先计算速度场，再计算温度场。在计算伴随热交换的流动问题时，这种方法效率比较高，其具体实现过程如下：

（1）取消 Solution Controls（求解控制）面板中能量方程的计算。

（2）在初始化时，用入口处气流温度与壁面温度的平均值初始化温度场。

（3）求解余下的方程直到收敛。

（4）重新打开 Solution Controls（求解控制）面板，取消流动方程计算，打开能量方程求解。

（5）求解能量方程直至获得收敛解。

周期性流动计算在计算过程中除了可以通过残差观察解的收敛过程外，还可以用温差比来观察求解过程是否收敛。在 Statistic Monitors（统计数据监视）面板中，选择 per/bulk-temp-ratio 作为监视变量即可。作为后处理的一部分，可以用流场中静温的等值线来观察计算结果。

7.11.4 浮力驱动流

浮力驱动流是流场中局部流体受热膨胀后，在浮力作用下发生的对流过程。浮力驱动流用 Rayleigh 数进行流态判断，即在 Rayleigh 数小于 10^8 时为浮力驱动的层流，在 Rayleigh 数处于 10^8 和 10^{10} 之间时为层流向湍流转捩的阶段，在大于 10^{10} 时为浮力驱动的湍流。

计算封闭空间内的对流过程时，密度场的计算是流场计算的关键。在 FLUENT 中可以用二种方法计算密度场：

- 一种是非定常方法，即流场的初始密度由初始压强和温度通过状态方程计算得到。在随后的计算中则用质量守恒关系计算密度场的变化。这种方法适用于温差比较大的流场计算。
- 另一种是采用 Boussinesq 模型的定常计算方法。所谓 Boussinesq 模型就是在计算中假设所有方程中的密度为常数，而只让动量方程对流项中的密度为变量进行计算。显然这种方法仅适用于小温差（因而密度差也很小）的流场计算。

计算参数的设置过程

浮力驱动流场计算的操作流程如下：

（1）在 Energy（能量）面板中激活能量方程计算。

（2）在 Operating Conditions（操作条件）面板中激活 Gravity（重力）选项，并设置 Gravitational Acceleration（重力加速度）的值和方向。

（3）如果计算中假设流场为理想不可压流场，则应在 Operating Conditions（操作条件）面板中正确设置 Operating Pressure（操作压强）的值。

（4）如果采用了 Boussinesq 模型，则应在 Operating Conditions（操作条件）面板中设定 Operating Temperature（操作温度），同时在 Materials（材料）面板中，激活 boussinesq 作为 Density（密度）的计算方法，并设定 Thermal Expansion Coefficient（热力学膨胀系数）。

（5）如果没有采用 Boussinesq 模型，则应该根据流场特点估算出平均密度，用这个密度值作为计算过程中的操作密度，并激活 Operating Conditions（操作条件）面板中的 Specified Operating Density（指定的操作密度）选项，然后在 Operating Density（操作密度）栏中填入操作密度的值。然后在 Materials（材料）面板中，将密度设置为温度的函数。

（6）在入口和出口边界上给定压强条件。

（7）在 Solution Controls（求解控制）面板上，Pressure（压强）的 Discretization（离散）方法中选用 Body Force Weighted（彻体力加权）或 Second Order（二阶精度）选项。在采用分离求解器时，应选用 PRESTO!作为压强的离散方法。

因为高 Rayleigh 数流动非常复杂，所以在实际计算中仍然可以象其他复杂流动计算那样，采用由简到繁的计算策略。对于高 Rayleigh 数计算大体上可以分三个阶段，即：

（1）低 Rayleigh 数计算。首先通过降低重力加速度的值，降低 Rayleigh 数，比如将重力加速度降低 2 个数量级，则 Rayleigh 数也可以随之降低 2 个数量级。因为在低 Rayleigh

数时，浮力驱动流为层流流动，因此很容易得到收敛解。

（2）一阶精度计算。以低 Rayleigh 数流场为初始流场，先用一阶精度格式进行计算。一阶精度格式的耗散作用很大，所以计算稳定性比较好，但是精度较差。

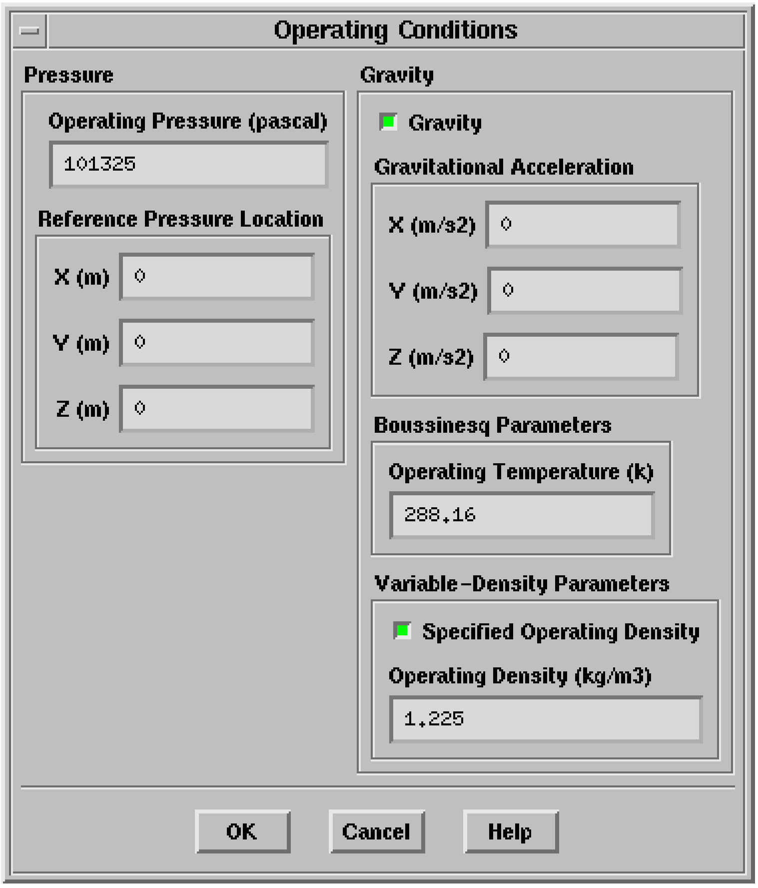


图 7-37 浮力驱动流场的设置

（3）高精度计算。在一阶精度计算结束后，再采用二阶精度格式继续计算，直到获得最终的流场。

浮力驱动流场计算的后处理过程与其他热交换计算没有什么区别，这里就不再重复。

7.12 本章小结

本章全面介绍了 FLUENT 中采用的物理模型的基本思想和具体操作，考虑到篇幅限制和读者的兴趣，将相关的理论推导过程省略了。感兴趣的读者可以通过查找相关专题的技术资料获得相关理论的知识。

第 8 章 边界条件

边界条件就是流场变量在计算边界上应该满足的数学物理条件。边界条件与初始条件一起并称为定解条件，只有在边界条件和初始条件确定后，流场的解才存在，并且是唯一的。FLUENT 的初始条件是在初始化过程中完成的，边界条件则需要单独进行设定。本章将详细讲述 FLUENT 中的边界条件的设定问题。

8.1 边界条件问题回顾

边界条件大致分为下列几类：

（1）流体进出口条件：包括压强入口、速度入口、质量入口、吸气风扇、入口通风、压强出口、压强远场、出口流动、出口通风和排气风扇等条件。

（2）壁面条件：包括固壁条件、对称轴（面）条件和周期性边界条件。

（3）内部单元分区：包括流体分区[1]和固体分区。

（4）内面边界条件：包括风扇、散热器、多孔介质阶跃和其他内部壁面边界条件。内面边界条件在单元边界面上设定，因而这些面没有厚度，只是对风扇、多孔介质膜等内部边界上流场变量发生阶跃的模型化处理。

8.1.1 边界条件面板

边界条件的设定是在边界条件面板中完成的，如图 8-1 所示。在读入算例文件后，通过下列菜单操作顺序点击可以启动边界条件面板：

Define -> Boundary Conditions...

1. 改变边界类型

在设定边界条件之前需要检查边界类型，比如名为 velocity-inlet-5 边界分区应该是 velocity-inlet（速度入口）类型，可是在检查前却被错误地设定为 pressure-inlet（压强入口）类型，此时就可以通过边界条件面板对边界类型进行修改，方法就是：

（1）在 Zone（分区）下面选择要进行修改的分区名，这里是 velocity-inlet-5。

（2）然后在 Type（类型）下面选择正确的边界，并在弹出的提示窗口中点击 OK 按钮确认上述选择。

[1] 多孔介质属于流体分区。

需要注意的是，周期性边界条件的类型不能用上述方式改变，多相流边界条件的设定也与上述步骤有区别，详情请看后面章节的介绍。

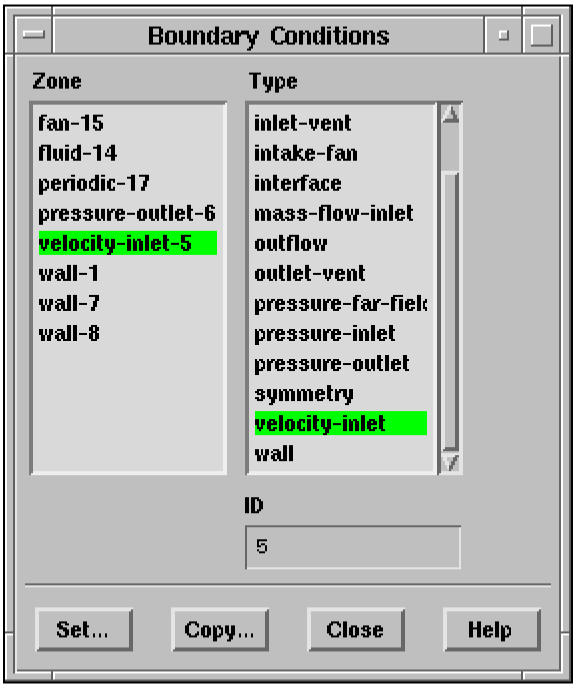


图 8-1 Boundary Conditions（边界条件）面板

2. 边界类型的分类

边界类型的改变是有一定限制的，不能随意进行修改。边界类型可以分成四个大类，所有边界类型都可以被划分到其中一个大类中。边界类型的改变只能在大类中进行，而分属不同大类的边界类型是不能互相替换的。这四个大类的分类情况见表 8-1。

表 8-1 边界类型的分类

大类	边界类型
面边界	轴边界、出口边界、质量流入口边界、压强远场条件、压强入口条件、压强出口条件、对称面（轴）条件、速度入口条件、壁面条件、入口通风条件、吸气风扇条件、出口通风条件、排气风扇条件
双面边界	风扇、多孔介质阶跃条件、散热器条件、壁面条件
周期性边界	周期条件
单元边界	流体、固体单元条件

3. 边界条件的设置

在边界条件面板中，点击下面的按钮 set（设置）按钮，或者双击分区面板下的分区名，将打开边界条件的设置面板，在这里可以对选定的边界分区进行边界条件的具体设置。

4. 边界条件的复制

除了直接设置边界条件外，如果还没有设定边界条件的分区与已经设定边界条件的某个分区的边界条件完全相同，则可以将现有的边界条件复制到新的边界分区中。

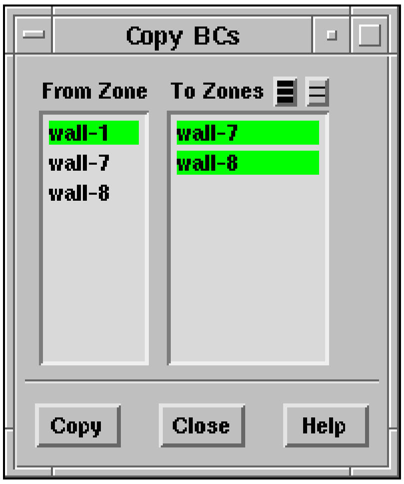


图 8-2 Copy BCs（边界条件复制）面板

边界条件复制的方法如下：

（1）在边界条件面板中点击下面的 Copy（复制）按钮，弹出图 8-2 所示的边界条件复制面板。

（2）在 From Zone（来源分区）下选定已经设置好边界条件的分区。

（3）在 To Zones（目标分区）下面选定目标分区。

（4）点击下方的 Copy（复制）按钮完成复制。

（5）点击 Close（关闭）按钮关闭边界条件复制面板。

注意内部边界和外部边界的边界条件不能互相复制，因为内部边界是双面边界，而外部边界是单面边界。

8.1.2 在图形显示窗口中选定边界分区

在边界分区数量较大时，从边界条件面板中选定边界分区不够直观，甚至会造成错误。为了避免这种错误，可以在图形显示窗口中用鼠标的右键直接指定边界分区。方法如下：

（1）顺序点击菜单 Display -> Grid 弹出图形显示窗口。

（2）在图形显示窗口中用鼠标右键指向目标边界，并点击完成选定。

此时在边界条件面板上，选定的边界分区的名称被反白显示。

8.1.3 修改边界设置

1. 编辑边界分区名称

用户如果不喜欢边界分区的名称，可以用下面的方法进行修改：

（1）在边界条件面板中，点击 Set（设置）按钮弹出边界条件设置面板。

（2）在 Zone Name（分区名称）下可以将边界分区名更改为自己喜欢的名称。

（3）点击 OK 按钮确认上述操作。

2. 定义非均匀边界条件

边界条件可以设定为常数形式，也可以设定为变量分布形式。常数形式就用上面的办法进行设定。变量分布形式可以用边界型函数文件进行设置，也可以用用户自定义函数（UDF）来进行设置。边界型函数文件将在 8.26 节中进行介绍，用户自定义函数（UDF）的使用方法将在本书的高级应用篇中进行介绍。

3. 定义瞬态边界条件

瞬态（transient）边界条件指随时间变化的边界条件。在 FLUENT 中，瞬态边界条件则进一步限定为仅随时间变化，不随空间变化的边界条件，即在任一时刻流场变量在边界上为一常数，但是在不同时刻，这个常数的值会发生变化。FLUENT 定义瞬态边界条件的方法有两种，其一是用边界型函数文件进行定义（请参见 8.26.5 小节内容），其二是用表格形式的数据进行定义。表格形式的数据格式为：

profile-name n_field n_data periodic?

field-name-1 field-name-2 field-name-3 .... field-name-n_field

v-1-1 v-2-1 ... ... v-n_field-1

v-1-2 v-2-2 ... ... v-n_field-2

.

.

.

v-1-n_data v-2-n_data ... ... v-n_field-n_data

其中 profile-name 是数据名称，n_field 为变量数，n_data 为每个变量所需输入的数据总数。periodic?用于定义边界是否为周期性边界，1 为是，0 为否。field-name 为变量名，集中在一行排列。下面是对变量的赋值，每一列数据对应赋值给上面相应的变量。

下面是一个例子：

sampletabprofile 2 3 1

time u

1 10

2 20

3 30

这个例子与前面提到的例子完全一样，只是书写格式有所不同。

需要注意的是，无论是采用型函数文件格式，还是采用表格形式的数据，文件内容必须采用小写字母，不能使用大写字母。另外，文件中数据必须使用国际单位制。使用表格形式的数据文件需要使用文字命令，其顺序及屏幕显示如下：

>file

/file>read-transient-table

transient file name []

然后在方括号后输入文件名即可。上述边界条件在保存后，可以重复读入到不同的算例中。

8.2 流动的入口和出口

8.2.1 边界条件分类

FLUENT 中的入口和出口边界包括下列十种形式：

（1）速度入口条件：在入口边界给定速度和其他标量属性的值。

（2）压强入口条件：在入口边界给定总压和其他标量变量的值。

（3）质量流入口条件：在计算可压缩流时，给定入口处的质量流量。因为不可压流的密度是常数，所以在计算不可压流时不必给定质量流条件，只要给定速度条件就可以确定质量流量。

（4）压强出口条件：用于在流场出口处给定静压和其他标量变量的值。在出口处定义出口（outlet）条件，而不是定义出流（outflow）条件，是因为前者在迭代过程中更容易收敛，特别是在出现回流的时候。

（5）压强远场条件：这种类型的边界条件用于给定可压缩流的自由流边界条件，即在给定自由流马赫数和静参数条件确定后，给定无限远处的压强条件。这种边界条件只能用于可压缩流计算。

（6）出流边界条件：如果在计算完成前无法确定压强和速度时，可以使用出流条件。这种边界条件适用于充分发展的流场，其做法是将除压强以外的所有流动参数的法向梯度都设为零。这种边界条件不适用于可压缩流。

（7）入口通风条件：这种边界条件的设置需要给定损失系数、流动方向、环境总压和总温。

（8）进气风扇条件：在假设入口处存在吸入式风扇的情况下，可以用这种边界条件设

定压强跳跃、流动方向、环境总压和总温。

（9）出口通风条件：在出口处给定损失系数、流动方向、环境总压和总温。

（10）排气风扇条件：在假设出口处存在排气风扇的情况下，给定出口处的压强跳跃和静压。

8.2.2 定义湍流参数

在流场的入口、出口和远场边界上，用户需要定义流场的湍流参数。在 FLUENT 中可以使用的湍流模型有很多种。在使用各种湍流模型时，哪些变量需要设定，哪些不需要设定以及如何给定这些变量的具体数值，都是经常困扰用户的问题。本小节只讨论在边界上设置均匀湍流参数的方法，湍流参数在边界上不是均匀分布的情况可以用型函数和 UDF（用户自定义函数）来定义，具体方法请参见相关章节的叙述。

在大多数情况下，湍流是在入口后面一段距离经过转捩形成的，因此在边界上设置均匀湍流条件是一种可以接受的选择。特别是在不知道湍流参量的分布规律时，在边界上采用均匀湍流条件可以简化模型的设置。在设置边界条件时，首先应该定性地对流动进行分析，以便边界条件的设置不违背物理规律。违背物理规律的参数设置往往导致错误的计算结果，甚至使计算发散而无法进行下去。

在 Turbulence Specification Method （湍流定义方法）下拉列表中，可以简单地用一个常数来定义湍流参数，即通过给定湍流强度、湍流粘度比、水力直径或湍流特征长在边界上的值来定义流场边界上的湍流。下面具体讨论这些湍流参数的含义，以保证在设置模型时不出现违背流动规律的错误设置：

（1）湍流强度（Turbulence Intensity）

湍流强度 I 的定义如下：

$$I = \frac{\sqrt{u'^2 + v'^2 + w'^2}}{u_{avg}} \qquad (8-1)$$

上式中 u' 、v' 和 w' 是速度脉动量，u_{avg} 是平均速度。

湍流强度小于 1％时，可以认为湍流强度是比较低的，而在湍流强度大于 10％时，则可以认为湍流强度是比较高的。在来流为层流时，湍流强度可以用绕流物体的几何特征粗略地估算出来。比如在模拟风洞试验的计算中，自由流的湍流强度可以用风洞的特征长度估计出来。在现代的低湍流度风洞中，自由流的湍流强度通常低于 0.05%。

内流问题进口处的湍流强度取决于上游流动状态。如果上游是没有充分发展的未受扰流动，则进口处可以使用低湍流强度。如果上游是充分发展的湍流，则进口处湍流强度可以达到几个百分点。如果管道中的流动是充分发展的湍流，则湍流强度可以用公式（8－2）计算得到，这个公式是从管流经验公式得到的：

$$I \equiv \frac{u'}{u_{avg}} = 0.16\left(\mathrm{Re}_{D_H}\right)^{-\frac{1}{8}} \qquad (8-2)$$

式中下标D_H是 Hydraulic Diameter（水力直径）的意思，即（8－2）式中的雷诺数是以水力直径为特征长度求出的。

（2）湍流的长度尺度与水力直径

湍流能量主要集中在大涡结构中，而湍流长度尺度l则是与大涡结构相关的物理量。在充分发展的管流中，因为漩涡尺度不可能大于管道直径，所以l是受到管道尺寸制约的几何量。湍流长度尺度l与管道物理尺寸L关系可以表示为：

$$l = 0.07L \qquad (8-3)$$

式中的比例因子 0.07 是充分发展管流中混合长的最大值，而L则是管道直径。在管道截面不是圆形时，L可以取为管道的水力直径。

湍流的特征长取决于对湍流发展具有决定性影响的几何尺度。在上面的讨论中，管道直径是决定湍流发展过程的唯一长度量。如果在流动中还存在其他对流动影响更大的物体，比如在管道中存在一个障碍物，而障碍物对湍流的发生和发展过程起着重要的干扰作用。在这种情况下，湍流特征长就应该取为障碍物的特征长度。

从上面的分析可知，虽然（8－3）式对于大多数管道流动是适用的，但并不是普遍适用的，在某些情况下可以进行调整。

在 FLUENT 中选择特征长L或湍流长度尺度l的方法如下：

1）对于充分发展的内流，可以用 Intensity and Hydraulic Diameter（湍流强度与水力直径）方法定义湍流，其中湍流特征长度就是 Hydraulic Diameter（水力直径）D_H。

2）对于导向叶片或分流板下游的流场，可以用 Intensity and Hydraulic Diameter（湍流强度与水力直径）定义湍流，并在 Hydrauli Diameter（水力直径）中将导向叶片或分流板的开口部分的长度L定义为特征长度。

3）如果进口处的流动为受到壁面限制且带有湍流边界层的流动，可以在 Intensity and Length Scale 面板中用边界层厚度δ_{99}通过公式$l = 0.4\delta_{99}$计算得到湍流长度尺度l。最后在 Turbulence Length Scale（湍流长度尺度）中输入l的值。

（2）湍流粘度比

湍流粘度比μ_t / μ与湍流雷诺数Re_t成正比。湍流雷诺数的定义为：

$$\mathrm{Re}_t = \frac{k^2}{\varepsilon \upsilon} \qquad (8-4)$$

Re_t在高雷诺数边界层、剪切层和充分发展的管道流动中的数值较大，其量级大约在

100 到 1000 之间。而在大多数外部流动的自由流边界上，μ_t / μ 的值很小。在典型情况下，μ_t / μ 的值在 1 到 10 之间。

用湍流粘度比定义流动时，可以使用 Turbulent viscosity Ratio（湍流粘度比）或 Intensity and Viscosity Ratio（湍流强度和粘度比）进行定义。前者适用于 Spalart-Allmaras 模型，后者适用于 $k-\varepsilon$ 模型、$k-\omega$ 模型和 RSM 模型。

（3）推导湍流变量时采用的关系式

为了从前面讲到的湍流强度 I 、湍流长度尺度 L 和湍流粘度比 μ_t / μ 求出其他湍流变量，必须采用几个经验关系式。在 FLUENT 中使用的经验关系式主要包括下面几种：

1）从湍流强度和长度尺度求出修正的湍流粘度

在使用 Spalart-Allmaras 模型时，可以用湍流强度 I 和长度尺度 l 求出修正的湍流粘度 $\tilde{\upsilon}$，具体公式如下：

$$\tilde{\upsilon} = \sqrt{\frac{3}{2}} u_{avg} I l \qquad (8-5)$$

在使用 FLUENT 时，如果在 Spalart-Allmaras 模型中选择 Intensity and Hydraulic Diameter（湍流强度与水力直径）选项，则修正的湍流粘度就用这个公式求出。其中的长度尺度 l 则用式（8－3）求出。

2）用湍流强度求出湍流动能

湍流动能 k 与湍流强度 I 的关系如下：

$$k = \frac{3}{2}\left(u_{avg} I\right)^2 \qquad (8-6)$$

式中 u_{avg} 为平均流速。

如果在使用 FLUENT 时没有直接输入湍流动能 k 和湍流耗散率 ε 的值，则可以使用 Intensity and Hydraulic Diameter（湍流强度与水力直径）、Intensity and Length Scale（湍流强度与长度尺度）或 Intensity and Viscosity Ratio（湍流强度与粘度比）等方法确定湍流动能，而确定的办法就是使用公式（8－6）。

3）用长度尺度求出湍流耗散率

长度尺度 l 与湍流耗散率之间的关系为：

$$\varepsilon = C_\mu^{3/4} \frac{k^{3/2}}{l} \qquad (8-7)$$

式中 C_μ 为湍流模型中的一个经验常数，其值约等于 0.09。

在没有直接输入湍流动能 k 和湍流耗散率 ε 的情况下，可以用 Intensity and Hydraulic

Diameter（湍流强度与水力直径）或 Intensity and Length Scale（湍流强度与长度尺度）等办法，利用上述公式确定湍流耗散率ε。

4）用湍流粘度比求出湍流耗散率

湍流耗散率ε与湍流粘度比μ_t/μ和湍流动能k的关系如下：

$$\varepsilon = \rho C_\mu \frac{k^2}{\mu}\left(\frac{\mu_t}{\mu}\right)^{-1} \qquad (8-8)$$

式中C_μ为湍流模型中的一个经验常数，其值约等于 0.09。

在没有直接输入湍流动能k和湍流耗散率ε的情况下，可以用 Intensity and Viscosity Ratio（湍流强度与粘度比）定义湍流变量，实际上就是利用上述公式算出湍流耗散率ε。

5）湍流衰减过程中湍流耗散率的计算

如果计算风洞阻尼网下游试验段中的流场，可以用下式求出湍流耗散率ε：

$$\varepsilon \approx \frac{\Delta k U_\infty}{L_\infty} \qquad (8-9)$$

式中Δk是湍流动能k的衰减量，比如可以设为入口处k值的 10%，U_∞是自由流速度，L_∞是自由流区域的长度。（8－9）式是对高雷诺数各向同性湍流衰减指数律的线性近似，其理论基础是衰减湍流中湍流动能k的方程：

$$U\frac{\partial k}{\partial x} = -\varepsilon \qquad (8-10)$$

如果用这种方法计算ε，还需要用（8－8）式检验计算结果，以保证湍流粘度比μ_t/μ不过大。虽然这种方法在 FLUENT 中没有使用，但是可以用这种方法估算出自由流中的湍流耗散率ε，然后再用（8－6）式确定k，最后在 Turbulence Specification Method（湍流定义方法）下拉列表中选择 K and Epsilon（k和ε）并k和ε的计算结果输入到相应的栏目中。

6）用长度尺度计算比耗散率

如果知道湍流长度尺度l，可以用下式确定ω：

$$\omega = \frac{k^{0.5}}{C_\mu^{0.25} l} \qquad (8-11)$$

式中C_μ和长度尺度l的取法与前面段落中所述相同。在使用 Intensity and Hydraulic Diameter（湍流强度与水力直径）或 Intensity and Length Scale（湍流强度与长度尺度）定

义湍流时，FLUENT 用的就是这种方法。

7）用湍流粘度比计算比耗散率

ω 的值还可以用 μ_t/μ 和 k 通过下式计算得出：

$$\omega = \rho\frac{k}{\mu}\left(\frac{\mu_t}{\mu}\right)^{-1} \qquad (8-12)$$

在使用 Intensity and Viscosity Ratio（湍流强度与粘度比）方法定义湍流时，FLUENT 就是使用上述关系式对湍流进行定义的。

8）用湍流动能定义雷诺应力分量

在使用 RSM（雷诺应力模型）时，如果用户没有在 Reynolds-Stess Specification Method（雷诺应力定义方法）的 Reynolds-Stress Components（雷诺应力分量）选项中直接定义雷诺应力的值，则雷诺应力的值将由给定的 k 值计算得出。假定湍流是各向同性的，即：

$$\overline{u'_i u'_j} = 0 \qquad (8-13)$$

且：

$$\overline{u'_\alpha u'_\alpha} = \frac{2}{3}k \qquad \text{（不对}\alpha\text{ 隐含求和）} \qquad (8-14)$$

如果用户在 Reynolds-Stress Specification Method（雷诺应力定义方法）下拉列表中选择 K or Turbulence Intensity（k 或湍流强度）时，FLUENT 就用这种方法定义湍流。

（4）在大涡模拟方法（LES）中定义进口湍流

在使用速度进口条件时，可以将湍流强度作为对 LES 进口速度场的扰动定义在边界条件中。在实际计算中，根据湍流强度求出的随机扰动速度分量与速度场叠加后形成 LES 算法边界上的、随机变化的速度场。

8.3 压强入口边界条件

压强入口边界条件用于定义流场入口处的压强及其他标量函数。这种边界条件既适用于可压流计算也适用于不可压流计算。通常用在入口处压强已知、而速度和流量未知时，就可以使用压强入口条件。压强入口边界条件还可以用于具有自由边界的流场计算。

8.3.1 压强入口边界的输入

在使用压强入口边界条件时需要输入下列参数：

（1）总压。

（2）总温。

（3）流动方向。
（4）静压。
（5）用于湍流计算的湍流参数。
（6）用于 P-1 模型、DTRM、DO 模型、“面到面”模型进行计算的辐射参数。
（7）用于组元计算的化学组元质量浓度。
（8）用于非预混或部分预混燃烧计算的混合物浓度和增量。
（9）用于预混或部分预混燃烧计算的过程变量。
（10）用于弥散相计算的弥散相边界条件（）。
（11）多相流边界条件（用于普通多相流计算）。

上述变量均在 Pressure Inlet（压强入口）面板中输入。在 Boundary Conditions（边界条件）面板中选择压强入口边界，然后点击 set（设置）按钮就可以进入压强入口条件的设置面板。

1. 压强与静水头

FLUENT 在计算压强时将压强与静水头 $\rho_0 gx$ 通过下式联系起来：

$$p'_s = p_s - \rho_0 gx \qquad (8-15)$$

对上式求导可得：

$$\frac{\partial p'_s}{\partial x} = \frac{\partial p_s}{\partial x} - \rho_0 g \qquad (8-16)$$

这样就可以将静水头引入彻体力项$(\rho - \rho_0)g$，并且在密度场均匀时可以将静水头从压强场计算中排除出去。因此在压强输入过程中应该忽略静水头变化所产生的影响，在计算结果中也不显示静水头的影响。详细内容请参考浮力驱动模型的内容。

2. 定义总压和总温

在 Pressure Inlet（压强入口）面板中的 Gauge Total Pressure（表总压）栏中输入总压的值，在 Total Temperature（总温）栏中输入总温。

这里总压输入仍然采用表压形式，即输入的总压值实际上是绝对总压与操作压强之差。不可压流中总压用伯努力方程定义，即：

$$p_0 = p_s + \frac{1}{2}\rho|\vec{v}|^2 \qquad (8-17)$$

在可压流中用满足等熵关系的伯努力方程定义，即：

$$p_0 = p_s\left(1+\frac{\gamma-1}{2}M^2\right)^{\frac{\gamma}{\gamma-1}} \qquad (8-18)$$

在公式（8－17）和（8－18）中，p_0 为实际总压，p_s 为实际静压，M 为马赫数，γ 为比热比。

在计算非对称旋转流时，（8－17）式中的 $\vec{v}$ 应该包括旋转分量。

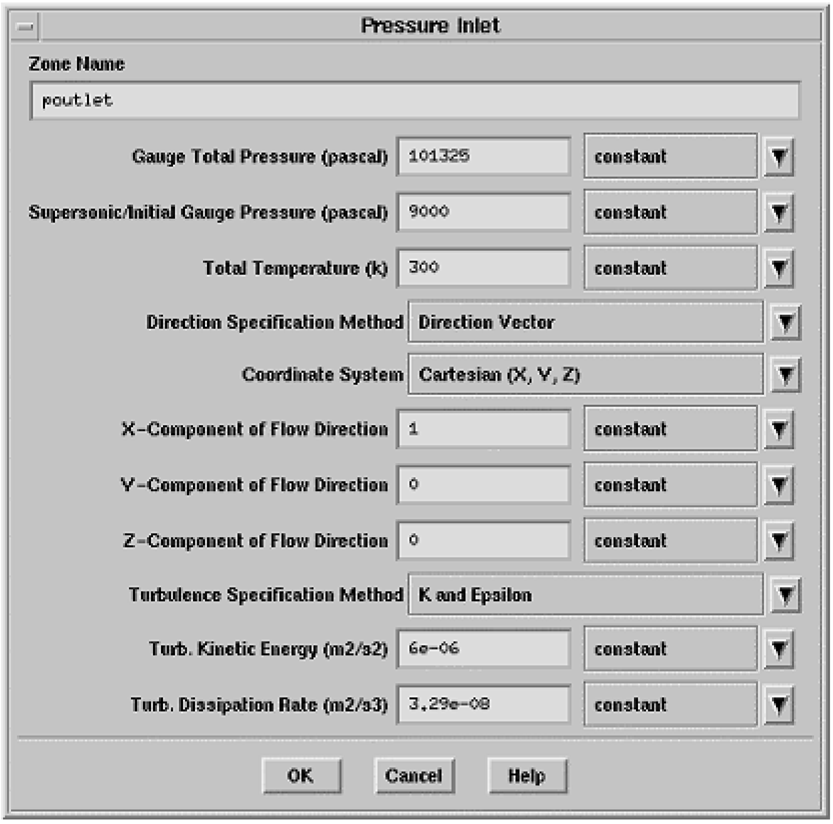


图 8-3 Pressure Inlet（压强入口）面板

如果相邻网格区域在移动，即在使用旋转坐标系、多重坐标系、混合平面或移动网格，并且同时采用分离求解器时，方程（8－17）中的速度和（8－18）中的马赫数是否与网格速度相关，取决于是否在 Solver（求解器）面板中开启了 Absolute（绝对）速度选项。如果开启了这个选项，则上述两方程中的速度和马赫数均为与网格速度无关的绝对速度，反之则为相对速度。在使用耦合求解器时，上述两方程中的速度和马赫数均与网格速度无关，为绝对速度。

3. 定义流动方向

在压强入口面板中可以用分量定义方式定义流动方向。在入口速度垂直于边界面时，也可以直接将流动方向定义为“垂直于边界”。具体设置过程中，既可以用直角坐标形式定义 x、y、z 三个方向的速度分量，也可以用柱坐标形式定义径向、切向和轴向三个方向的速度分量。

对于移动区域问题，如果选用分离求解器，则流动方向究竟是独立于网格速度，还是相对于网格速度进行定义，取决于 Solver（求解器）面板中是否开启了 Absolute（绝对速度）选项。如果选用耦合求解器，则流动方向必须用绝对速度进行定义。

定义流动方向的步骤如下：

（1）选择流动方向的定义方法，即在 Direction Specification Method（方向定义方法）下拉列表中选择 Direction Vector（方向矢量）或 Normal to Boundary（垂直于边界）。

（2）如果选择了 Normal to Boundary（垂直于边界），并且准备计算的是轴对称旋转流问题，则要在 Tangential-Component of Flow Direction（流动方向的分量）中输入适当的数值。如果选择的是 Normal to Boundary（垂直于边界），同时采用了直角坐标，并且不是轴对称旋转流动问题，则无需输入任何参数。

（3）如果在前面步骤中选择的是 Direction Vector（方向矢量），并且所计算的问题是三维问题，随后就需要选择矢量分量的坐标系形式，即在 Coordinate System（坐标系）下拉列表中选择 Cartesian（X,Y,Z）（直角坐标）、Cylindrical（Radial,Tangential,Axial）（柱坐标）或 Local Cylindrical（Radial,Tangential,Axial）（局部柱坐标）。

这里选用的直角坐标系就是网格划分时使用的直角坐标系。柱坐标系中的轴向、径向和切向分量则取决于下列坐标系：

1）在只包含一个单独网格区域的计算中，坐标系由 Fluid（流体）面板中定义的旋转轴和原点定义。

2）在包含多个区域的计算中（即计算中采用了多个参考系或使用了滑动网格），坐标系由靠近入口处的流体或固体区域中的旋转轴定义，这个旋转轴也是预先在 Fluid（流体）或 Solid（固体）面板中定义的。

对于上面定义的柱坐标系，径向速度以从旋转轴向外为正，轴向速度以旋转轴矢量的正向为正，切向速度则按照右手法则由径向和轴向速度矢量确定其正的方向。

3）局部柱坐标系允许用户为流场入口专门定义一个坐标系。如果用户选择了局部柱坐标系的选项，就可以在 Pressure Inlet（压强入口）面板上定义局部柱坐标系。如果计算中有多个流场入口，并且每个入口都有不同的转动轴，则局部柱坐标系就非常有用。

（4）如果在（1）中选择的是定义方向矢量，则可以用下列方法定义方向矢量的分量：

1）如果计算中的几何构型是二维非轴对称的，或者在（3）中选择的是输入直角坐标分量，在这里就直接输入流动方向矢量在 X、Y 和 Z 三个方向分量的值。

2）如果几何构型是二维轴对称的，或者在（3）中选择的是输入柱坐标分量，在这里就直接输入流动方向的轴向、径向和切向分量的值。

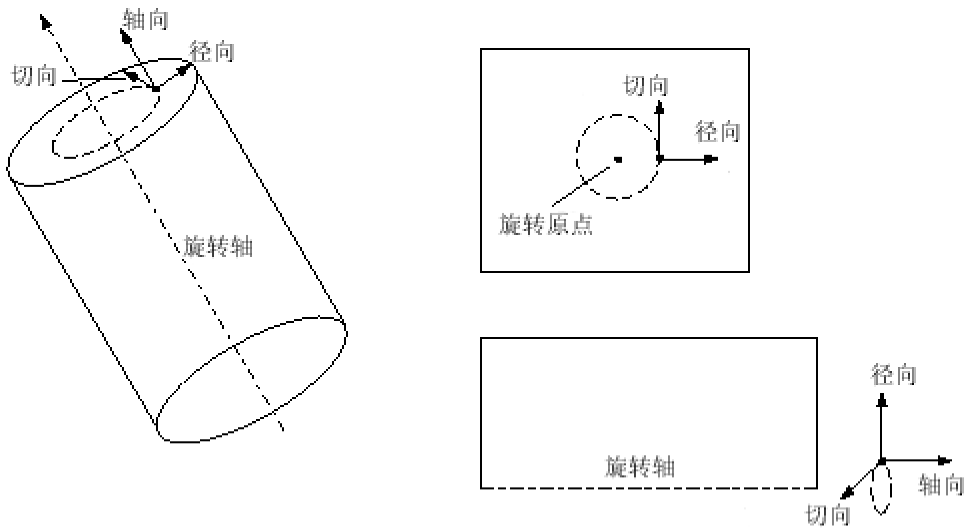


图 8-4 三维、二维和轴对称区域中，柱坐标系速度分量的定义

3）如果用户选择的是局部柱坐标，则输入流动方向矢量的轴向、径向和切向的分量数据，然后定义轴向矢量的 X、Y 和 Z 的分量值，以及转轴原点的 X、Y 和 Z 向的坐标值。

4. 定义静压

静压在 FLUENT 中被称为 Supersonic/Initial Gauge Pressure（超音速/初始表压），如果入口流动是超音速的，或者用户准备用压强入口边界条件进行计算的初始化工作，则必须定义静压。

在流场为亚音速流场时，FLUENT 将忽略 Supersonic/Initial Gauge Pressure（超音速/初始表压）的输入数据，而用驻点参数求出静压。然而如果使用压强入口条件进行流场的初始化，则需要同时输入静压和驻点参数，以便可以用（8－18）式或者（8－17）式计算出初始流场。因为上述两个公式中使用了马赫数和速度，所以在定义边界条件时需要给出合理的入口马赫数（可压流）或入口速度（不可压流）。

如果在计算中使用了湍流模型、辐射模型、燃烧模型等物理模型，还需要定义与这些模型相关的参数，详细的定义方法请参见相关模型的内容。

8.3.2 压强入口边界的缺省设置

压强入口边界的缺省设置采用国际单位制，其中所有压强项仍然采用表压形式，具体如下：

表 8-2 压强入口边界的缺省设置

流动参数	缺省值
表总压	0
超音速/初始表压	0
总温	300
流动方向矢量的 X 分量	1
流动方向矢量的 Y 分量	0
流动方向矢量的 Z 分量	0
湍流动能	1
湍流耗散率	1

8.3.3 压强入口边界的计算流程

FLUENT 处理压强入口边界条件的方式是在假定没有流动损失的前提下，由驻点条件计算出入口条件。对于不可压流动，是从（8－17）式推导出入口边界条件。对于可压流，则用（8－18）式推导出入口边界条件。

1. 不可压流压强入口边界条件的计算

在流体流入压强边界时，FLUENT 将用户输入的压强作为入口平面的总压 p_0。在不可压流动中，可以用入口总压 p_0 和静压 p_s 通过（8－17）式计算出入口速度。在设定了流动方向矢量和速度的绝对值后，可以求出速度的分量。

对于不可压流，入口平面上的密度既可以直接设置为常数，也可以作为温度、组元质量浓度的函数计算得出。

如果流体流出压强入口边界，则设定的总压被作为静压使用。在流出得流体为不可压流时，输入的总温被作为静温使用。

2. 可压流压强入口边界的计算

在可压流计算中，压强入口边界的总压、静压和速度可以用理想气体的等熵关系联系起来。用户在入口边界上输入的总压 p'_0 和相邻单元上的静压 p'_s 用下列等熵关系联系在一起：

$$\frac{p'_0+p_{op}}{p'_s+p_{op}}=\left(1+\frac{\gamma-1}{2}M^2\right)^{\frac{\gamma}{\gamma-1}} \qquad (8\text{-}19)$$

其中马赫数定义为：

$$M=\frac{v}{c}=\frac{v}{\sqrt{\gamma RT_s}} \qquad (8\text{-}20)$$

c 为音速，γ 为比热比。式（8-19）中 p_{op} 为操作压强，因为输入的边界条件是以操作压强为参照的相对压强，所以实际压强的值应该是输入压强与操作压强之和。在给定 p'_0 和 p'_s 后，可以用（8-19）和（8-20）式计算得出入口平面上的流速，然后就可以用方向矢量分量求出各坐标方向上的速度分量。

对于可压流，入口平面上的密度值可以根据理想气体的状态方程计算得到：

$$\rho=\frac{p'_s+p_{op}}{RT_s}$$

式中的气体常数 R 可以用边界条件中组分的质量浓度 Y_i 计算得出。入口静温 T_s 可以用总温算出，具体公式如下：

$$\frac{T_0}{T_s}=1+\frac{\gamma-1}{2}M^2$$

8.4 速度入口边界条件

速度入口边界条件用入口处流场速度及相关流动变量作为边界条件。在速度入口边界条件中，流场入口边界的驻点参数是不固定的。为了满足入口处的速度条件，驻点参数将在一定的范围内波动。

需要注意的是，因为这种条件中允许驻点参数任意浮动，所以速度入口条件仅适用于不可压流，如果用于可压流则可能导致非物理解。同时还要注意的是，不要让速度入口条件过于靠近入口内侧的固体障碍物，这样会使驻点参数的不均匀程度大大增加。

在特殊情况下，可以在流场出口处也使用速度入口条件。这种情况下，必须保证流场在总体上满足连续性条件。

8.4.1 速度入口边界条件的输入

在速度入口条件中需要输入下列参数：

（1）速度值及方向，或速度分量。
（2）二维轴对称问题中的旋转速度。
（3）用于能量计算的温度值。
（4）使用耦合求解器时的出流表压。
（5）湍流计算中的湍流参数。
（6）采用 P-1 模型、DTRM、DO 模型或“面到面”模型时的辐射参数。
（7）组元计算中的化学组元质量浓度。
（8）非预混模型或部分预混模型燃烧计算中的混合物浓度及增量。
（9）预混模型或部分预混模型燃烧计算中的过程变量。
（10）弥散相计算中弥散相的边界条件。
（11）多相流计算中的多相流边界条件。

所有数值均在 Velocity Inlet（速度入口）面板上输入，如图 8-5。在 Boundary Condition（边界条件）面板中选择定义速度入口的边界，点击 set（设置）按钮就可以打开 Velocity Inlet（速度入口）面板。

1. 定义速度

因为速度为矢量，所以定义速度包括定义速度的大小和方向两个内容。在 FLUENT 中定义速度的方式有三种：第一种是将速度看作速度的绝对值与一个单位方向矢量的乘积，然后通过定义速度的绝对值和方向矢量分量定义速度；第二种是将速度看作在三个坐标方向上的分量的矢量和，然后通过分别给定速度三个分量大小定义速度；第三种是假定速度是垂直于边界面的（因此方向已知），然后只要给定速度的绝对值就可以定义速度。

FLUENT 中既可以使用相对速度，也可以使用绝对速度定义速度。在临近网格是静止网格时，这两种速度定义方式是完全等价的。如果靠近速度入口的单元区域是移动的，用户就可以根据需要从二种方式中选择一种。而在定义速度过程中就一定要注意究竟用的是相对速度定义还是相对速度定义。

与速度矢量定义的另一个相关因素是坐标系的形式，FLUENT 提供直角坐标系、柱坐标系和局部柱坐标系三种坐标系形式。在定义速度的过程需要注意究竟使用的是哪种坐标系，然后再进行分量计算，最后得到速度的定义所需要的所有数据。

在计算轴对称旋转流（axsiymetric swirl）问题时，要注意除了二维平面内的轴向速度和径向速度外，还有一个垂直于屏幕方向的切向旋转速度需要定义。

（1）定义速度的方法在下拉菜单 Velocity Specification Method（速度定义方法）中选择，选项包括 Magnitude and Direction（给定速度值与速度方向）、Components（给定速度分量）和 Magnitude,Normal to Boundary（给定速度值且假定速度垂直于边界）。

（2）如果靠近速度入口边界的是移动网格，则在 Reference Frame（参考坐标系）下拉列表中，选择 Relative to Adjacent Cell Zone（以相邻网格为参照的相对速度）或 Absolute（绝对速度）选项确定所定义的速度是相对速度还是绝对速度。

（3）坐标系是在 Coordinate System（坐标系）下拉列表中，通过选择 Cartesian (X, Y, Z)

（直角坐标系）、Cylindrical (Radial, Tangential, Axial)（柱坐标系）或 Local Cylindrical (Radial, Tangential, Axial)（局部柱坐标系）来确定的。

（4）在确定了速度定义方法、是否使用相对速度、坐标系形式三个选择后，就可以输入栏中分别输入相应的数据即可完成速度定义。

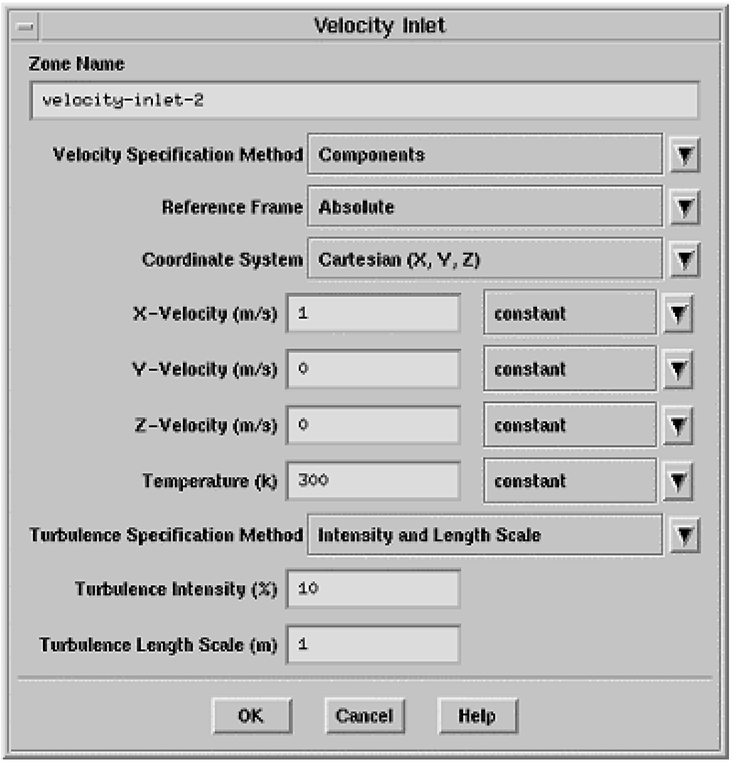


图 8-5 Velocity Inlet（速度入口）面板

在计算模型是轴对称带旋转流动时，除了可以定义旋转速度，还可以定义旋转角速度。类似地，如果选择了柱坐标系或局部柱坐标系，则除了可以定义切向速度，还可以定义入口处的角速度。将角速度看作矢量，则其定义与速度矢量定义是类似的。

2. 定义温度和出口表压

如果计算中包含了能量方程，则需要在入口速度边界处需要给定静温。

如果采用耦合求解器，还可以在速度入口边界上定义出流表压（Outflow Gauge Pressure）。如果在计算过程中速度入口边界上出现回流，则那个面就被作为压强出口边界处理，其中使用的压强就是在这里定义的出流表压。

除了上面的参数外，在速度入口边界条件中还可以定义湍流参数、组元质量浓度、辐射参数、非预混燃烧参数、预混燃烧边界条件、弥散相边界条件、多相流边界条件等参数，具体方法见后面与这些模型相关的章节。

8.4.2 速度入口边界上的缺省值

速度入口边界上的缺省设置见下表：

表 8-3 速度入口边界的缺省设置

参数	数值
温度（Temperature）	300
速度（Velocity Magnitude）	0
流动方向的 X 分量	1
流动方向的 Y 分量	0
流动方向的 Z 分量	0
X 方向的速度	0
Y 方向的速度	0
Z 方向的速度	0
湍流动能	1
湍流耗散率	1
出口流动表压	0

8.4.3 速度入口边界上流动参数的计算

在采用速度入口边界条件时，可以给定的速度求出质量流量、动量通量和能量通量。在带组元的流动计算中，还可以计算出组元质量浓度的通量。计算这些变量的方法如下：

1. 流量计算

在使用速度入口边界条件时，FLUENT 利用边界条件中的速度和温度等标量参数计算质量流量、动量通量、能量通量和化学组元。

质量流量的计算公式为：

$$\dot{m} = \int \rho \vec{v} \cdot d\vec{A} \quad (8-21)$$

上式中速度与面积单元之间进行的是点乘计算，所以实际上只有法向速度分量对质量流计算有贡献。

计算中有时会在流场出口处使用速度入口条件定义出口处的流量，比如在流场出口处的流量如果是已知的，就可以使用这种方法。当然在这类计算中必须始终保证流场在总体上满足连续性条件。

在分离算法中，如果某个流场出口处使用了速度入口条件，则 FLUENT 就用这个边界条件中定义的速度法向分量计算出口流量，而其他流场参数则使用出口边界上游单元的值

作为边界值。

在耦合算法中，如果某个边界面为流体流出流场的表面，则这个面上将使用压强出口条件，即这个面上的压强是在出流表压场中定义的压强值。

2. 密度计算

入口平面上的密度可以用二种方法得出：

（1）在流动为不可压流时，可以认为密度是常数，从而可以在计算开始前设定。

（2）在可压流中，可以将密度作为温度、压强的函数通过计算得出。在考虑组元的计算中，密度还是组元质量浓度的函数。在带组元的计算中，组元质量浓度在边界上的值是作为入口边界条件预先给定的。

3. 动量通量、能量通量和组元质量浓度通量的计算

在使用 SIMPLE 算法时，压强场在初始化过程中是人为给定的，而在随后的迭代计算过程中压强场被不断刷新。在获得上述压强场数值解的前提下，可以在速度入口边界上通过计算得到动量通量和能量通量的值。

在带组元计算的速度入口条件中可以定义组元在边界上的质量浓度。利用已知的组元质量浓度和速度就可以算出组元质量浓度的通量。

从上述计算过程中我们可以发现，在使用速度入口条件时，边界上的总压和静压都是随迭代计算的发展不断变化的。

8.5 质量流入口边界条件

在已知流场入口处的流量时，可以通过定义质量流量或者质量通量分布的形式定义边界条件。这样定义的边界条件叫做质量流入口边界条件。在质量流量被设定的情况下，总压将随流场内部压强场的变化而变化。

如果流场在入口处的主要流动特征是质量流量保持不变，则适合采用质量流入口条件。但是因为流场入口总压的变化将直接影响计算的稳定性，所以在计算中应该尽量避免在流场的主要入口处使用质量流条件。比如在带横向喷流的管道计算中，管道进口处应该尽量避免使用质量流条件，而在横向喷流的进口处则可以使用质量流条件。

在不可压流计算中不需要使用质量流入口条件，这是因为在不可压流中密度为常数，所以采用速度入口条件就可以确定质量流量，因此就没有必要再使用质量流入口条件。

8.5.1 输入质量流入口条件

在采用质量流入口条件时，需要输入下列参数：

（1）质量流量、质量通量，或混合面模型计算时的平均质量通量。
（2）总温。
（3）静压。
（4）流动方向。
（5）在湍流计算中输入湍流参数。
（6）在使用辐射模型时输入辐射参数。
（7）在带组元计算中输入化学组元质量浓度。
（8）在非预混和部分预混燃烧计算中输入混合物浓度与增量。

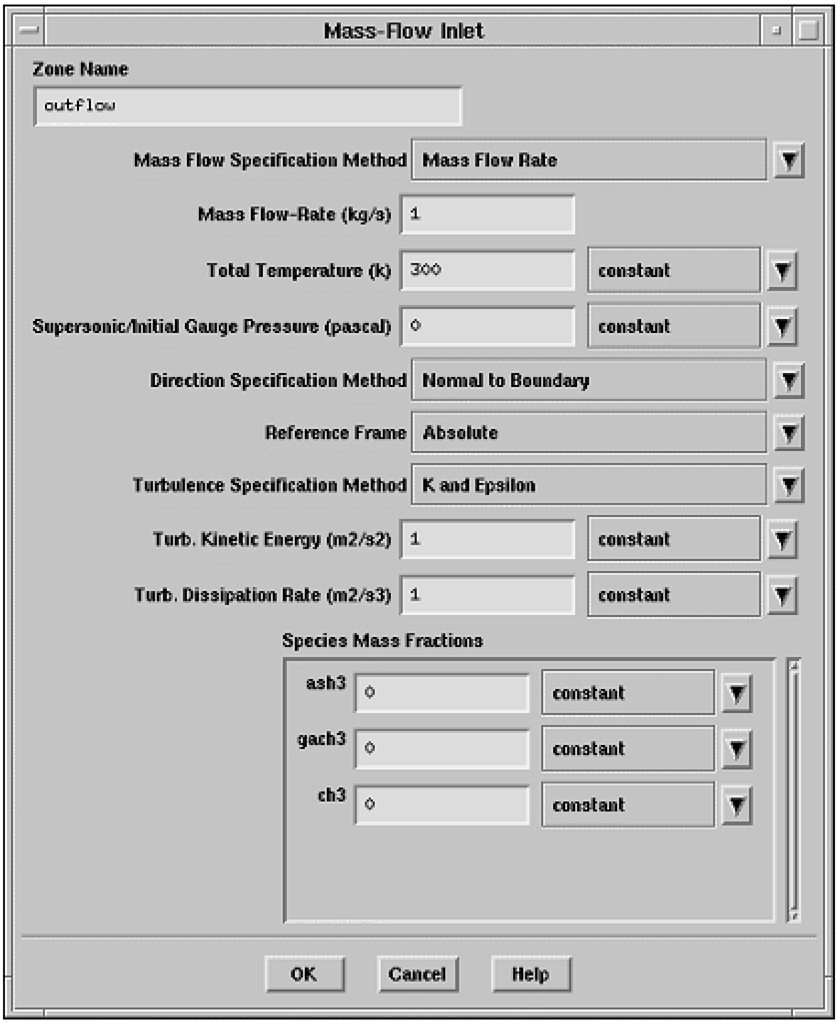


图 8-6 Mass-Flow Inlet（质量流入口）面板

（9）在预混或部分预混燃烧计算中输入过程变量。
（10）在弥散相模型计算中设定弥散相边界条件。
（11）在多相流计算中定义多相流边界条件。

所有输入的值都在质量流入口面板（Mass-Flow Inlet）中输入。在 Boundary Conditions（边界条件）面板中选择入口边界，然后点击 set（设置）按钮既可进入这个面板。

1. 定义质量流量或质量通量

可以在入口边界上定义质量流量，对于随时间变化的质量流量，则可以定义平均流量。如果在边界条件中定义的是质量流量，FLUENT 会自动将其转换为质量通量形式。当然也可以采用型函数或用户自定义函数的形式直接定义质量通量。

质量流量或质量通量的输入方式如下：

（1）质量流量的定义方法在质量流定义方法（Mass Flow Specification Method）下拉列表中进行选择，其中包括质量流量（Mass Flow Rate）、质量通量（Mass Flux），或平均质量通量三个选项。

（2）定义质量流量（Mass Flow Rate）为 FLUENT 中的缺省设置。这种定义方法就是通过给定质量流量来定义边界条件。需要注意的是，在轴对称流场计算中，输入的流量应该是整个截面（角度范围为 2π ）上的流量，而不是 1 弧度切片上的流量。

如果选择质量通量（Mass Flux）作为输入方法，则在质量通量栏中输入质量通量数据。在轴对称条件下，质量通量是 1 弧度切片上的质量通量。在用质量通量设置边界条件时，允许使用型函数文件定义通量分布，因而可以更准确地表示质量流在入口处的分布。

如果选择平均质量通量（Mass Flux with Average Mass Flux）作为输入方法，则要在质量通量和平均质量通量两个栏目中给定质量通量和平均质量通量的值。其中质量通量可以用常数形式定义，也可以用型函数定义其在网格上的分布，而平均质量通量则是所有网格上质量通量的积分。在轴对称流场计算中，这个通量是指 1 弧度切片上的质量通量。

如上所述，平均质量通量是与质量通量同时设定的。这点在某些情况下是非常有用的，比如已知入口处的质量通量分布，同时还知道总的质量流量等于 5。此时可以通过调入型函数文件定义质量通量在边界面上的分布，用平均质量通量定义质量流量。如果质量通量在边界面上的积分与已知的质量流量不相等，比如等于 4.7。此时如设定平均质量通量为 5，则 FLUENT 可以在保持质量通量分布形态不变的情况下，通过调整质量通量的值，使最终得到的质量流量等于 5。

平均质量通量的定义方法还可以用在混合面模型中。如果质量流入口边界是一个混合面，则不需要设定质量通量或流量，只要保持质量流量的值为缺省的 1 就可以。在算例设置的过程中创建混合平面时，FLUENT 会自动在质量流入口处选择平均质量通量法，并通过对上游流场的积分获得平均质量通量。这样可以保证质量在上下游是守恒的。

2. 定义总温和静压

直接在质量流入口面板上总温（Total Temperature）一栏中输入总温的值即可。

如果入口流动是超音速流动，或者计算是基于压强入口边界条件进行的，则需要在 Supersonic/Initial Gauge Pressure（超音速/初始表压）一栏中输入静压的值。在流动是亚音速时，这一栏中的输入内容将被 FLUENT 忽略。如果流场的初始化过程是基于质量流入口

条件的，则可以根据（8－18）式用输入的静压计算初始总压。需要再次提醒的是，流场中实际的静压值应该等于操作压强与输入静压之和。

3. 定义流动方向

在 Direction Specification Method（方向定义方法）中可以选择质量流入口边界上流动方向的定义方式。在流动方向与边界面不垂直时可以选择 Diretion Vector（方向矢量）方式。在流动方向与边界垂直时，则可以直接定义为 Normal to Boundary（垂直于边界）。

如果与入口相邻的网格单元是移动的，则可以在 Reference Frame（参考坐标系）下拉列表中通过选择 Absolute（绝对坐标系）或 Relative（相对坐标系）来指定定义方向矢量的坐标系形式。如果相邻网格单元不是移动的，则两种定义方式是等价的，因而无需进行任何选择。

与前面几种边界条件类似，在质量流入口条件中也可以定义湍流参数、辐射参数、组元质量浓度、非预混燃烧模型参数、预混燃烧边界条件、弥散相边界条件、多相流边界条件等计算边界条件，具体方法请参考相关章节的叙述。

8.5.2 质量流入口边界条件的缺省设置

质量流入口边界条件的缺省设置见表 8-3。

表 8-4 质量流入口边界条件的缺省设置

参数	数值
质量流量	1
总温	300
超音速/初始表压	0
流动方向的 X 向分量	1
流动方向的 Y 向分量	0
流动方向的 Z 向分量	0
湍流动能	1
湍流扩散率	1

8.5.3 质量流入口边界条件的计算流程

当在入口处采用质量流边界条件时，入口处每个单元表面上的速度值都可以用质量流条件计算出来，计算出的速度则用于计算其他变量在边界上的通量。每次迭代计算得到的速度值都在不断调整，以便在边界上满足质量流入口条件。计算速度需要用到边界条件中的质量流量、流动方向、静压和总压的数据。

定义质量流量的方法有两种，第一种是在入口处定义质量流量 $\dot{m}$ ，即单位时间通过入口边界的流量；第二种是定义质量通量 ρv_n ，即单位时间单位面积上通过入口边界的流量。这两个变量的关系如下：

$$\rho v_n = \frac{\dot{m}}{A} \tag{8-22}$$

如果选择直接定义流量通量，则流量通量沿边界面分布的变化可以用型函数文件或用户自定义函数来定义。如果还使用了平均质量通量方法，则如前所述，质量通量型函数将根据平均质量通量进行修正，以保证在边界上满足平均质量流量条件。

如果在给定平面上设定了 ρv_n 的值，为了计算法向速度的值，还需要定义边界上的密度 ρ 。密度的计算方式取决于所计算的流体是不是理想气体。如果流体是理想气体，则可以用静压 p 和静温 T 通过状态方程得到密度的值，状态方程为：

$$p = \rho RT \tag{8-23}$$

在入口流动为超音速时，静压的值等于在边界条件中设定的值。如果入口为亚音速，则静压的值是用入口边界相邻网格内的压强外插得到的。

静温是用总焓求出的，而总焓则是总温的函数，总温则是在边界条件中设定的，这几个参数之间的关系如下：

$$h_0(T_0) = h(T) + \frac{1}{2}v^2 \tag{8-24}$$

在计算不可压流时，入口处密度可以看作常数，或者是温度和组元浓度的函数。速度则用流量计算得到。

所有流动参数在入口边界上的通量都可以直接用法向速度计算得出。比如，质量流量等于 ρv_n ，湍流动能的通量等于 $\rho k v_n$ 等等。在有限体积法的计算中，这些通量的值可以作为边界条件，用于求解守恒型方程。

8.6 通风入口边界条件

通风入口边界条件通过给定损失系数、流动方向、环境压强和温度定义流场入口处的边界条件。通风入口边界条件中需要输入的参数如下：

（1）总压。

（2）总温。

（3）流动方向。

（4）静压。

（5）湍流参数。
（6）辐射参数。
（7）化学组元浓度。
（8）混合物浓度和增量。
（9）预混或部分预混燃烧计算中的过程变量。
（10）弥散相计算中的弥散相边界条件。
（11）多相流计算中的多相流边界条件。
（12）损失系数。

所有参数均在 Inlet Vent（通风入口）面板中设置，这个面板也是在 Boundary Conditions（边界条件）面板中启动的。

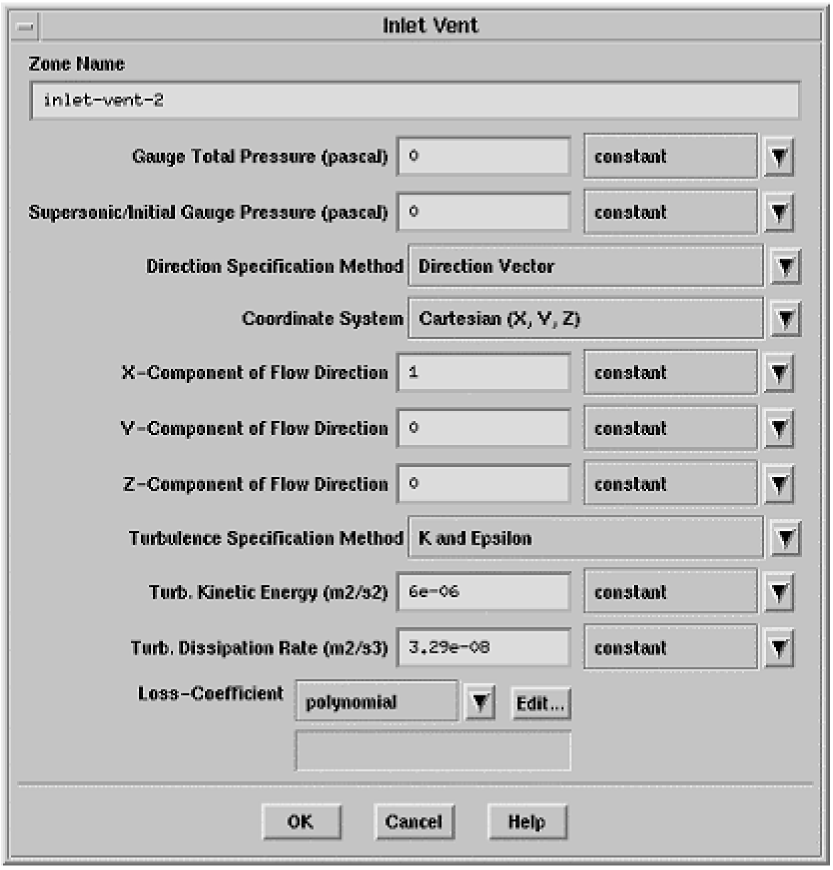


图 8-7 Inlet Vent（通风入口）面板

通风入口边界平面上的压力降正比于流体的动压头与一个损失系数的乘积，即压力降为入口边界上法向速度的函数：

$$\Delta p = k_L \frac{1}{2} \rho v_n^{\ 2} \qquad (8\text{-}25)$$

式中 k_L 为无量纲损失系数。在 Inlet Vent（通风入口）面板上，损失系数的定义方式在 Loss-Coefficient（损失系数）下拉列表中定义，其中包括 constant （常数)、polynomial（多项式拟合)、piecewise - linear（阶梯函数）或 piecewise-polynomial（分段多项式）四种定义形式。其他参数的设置方法与压强边界条件中的相同，这里不再重复。

8.7 进气风扇边界条件

在流场入口处为进气风扇时，可以用压强跃升、流动方向、环境压强和温度等参数的集合作为进气风扇的简化模型并作为入口边界条件。

进气风扇边界条件的输入参数如下：

（1）总压。

（2）总温。

（3）流动方向。

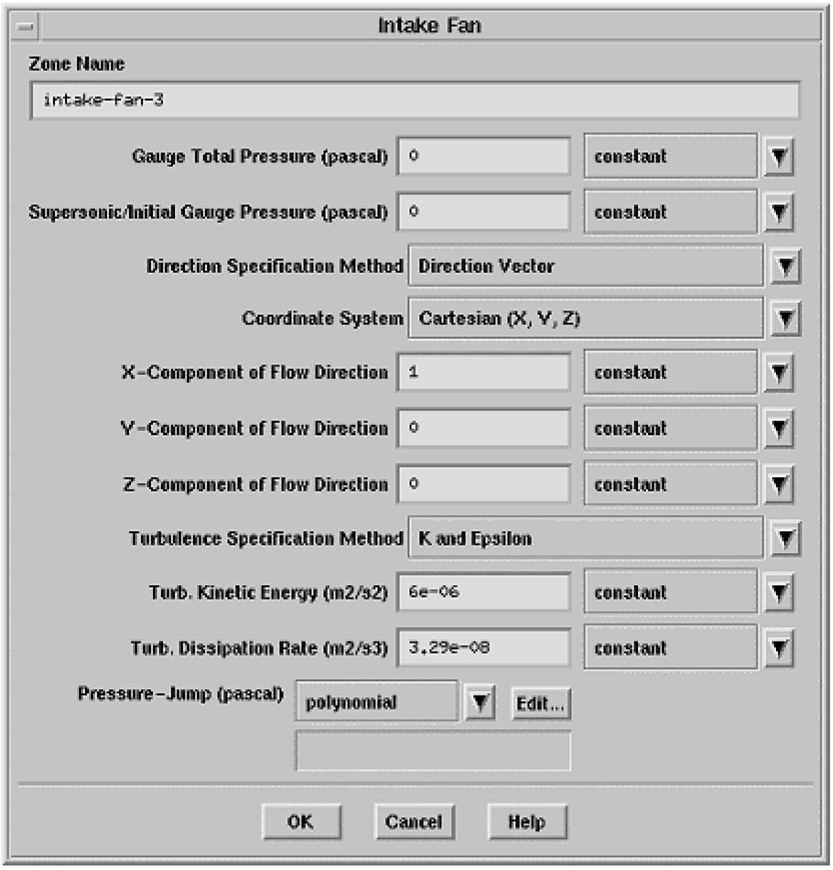


图 8-8 Intake Fan（进气风扇）面板

（4）静压。
（5）湍流计算中输入湍流参数。
（6）辐射计算中输入辐射参数。
（7）组元计算中输入化学组元的质量浓度。
（8）非预混或部分预混燃烧计算中的混合物浓度和增量。
（9）预混或部分预混燃烧计算中的过程变量。
（10）弥散相计算中的弥散相边界条件。
（11）多相流计算中的多相流边界条件。
（12）压强跳跃。

所有参数均在进气风扇（Intake Fan）面板输入，这个面板也是在 Boundary Conditions（边界条件）面板中开启。

在进气风扇边界条件中，风扇所在空间实际上被简化为一个无限薄平面。压强的跃升则被定义为通过风扇的流体速度的函数。在有回流时，风扇边界则被简化为损失系数等于 1 的出气风扇。

压强跃升的定义方法是在 Pressure-Jump（压强跃升）下拉列表中进行选择的，其中可供选择的选项也包括 constant （常数）、polynomial（多项式拟合）、piecewise - linear（阶梯函数）或 piecewise-polynomial（分段多项式）四种定义形式。

8.8 压强出口边界条件

压强出口边界条件在流场出口边界上定义静压，而静压的值仅在流场为亚音速时使用。如果在出口边界上流场达到超音速，则边界上的压强将从流场内部通过插值得到。其他流场变量均从流场内部插值获得。

在压强出口边界上还需要定义“回流（backflow）”条件。回流条件是在压强出口边界上出现回流时使用的边界条件。推荐使用真实流场中的数据做回流条件，这样计算将更容易收敛。

FLUENT 在压强出口边界条件上可以使用径向平衡条件，同时可以给定预期的流量。

8.8.1 压强出口边界的输入参数

压强出口边界的输入参数如下：

（1）静压。
（2）回流条件，其中包括：

- 能量计算中的总温。
- 回流方向定义方法。
- 湍流计算中的湍流参数。

- 组元计算中的化学组元质量浓度。
- 非预混或部分预混燃烧计算中的混合物浓度和增量。
- 预混或部分预混燃烧计算中的过程变量。
- 多相流计算中的多相流边界条件。

（3）辐射计算中的辐射参数。

（4）弥散相计算中的弥散相边界条件。

所有参数均在 Pressure Outlet（压强出口）面板中输入，压强出口面板在边界条件面板中开启。

1. 定义静压

在压强出口面板的 Gauge Pressure（表压）一栏中填入静压的值。在流动为亚音速时会用到这个值，如果在出口边界附近流动转变为超音速，则压强的值是从上游流场中外插得到的。再次提醒注意的是静压的值是以操作压强为参照的相对值。

在 FLUENT 中还可以使用径向平衡出口边界条件。在压强出口面板中选中 Radial Equilibrium Pressure Distribution（径向平衡压强分布），就可以启用这项功能。径向平衡指的是在出口平面上径向压强梯度与离心力的平衡关系。这种边界条件的设定方法只需要设定最小半径处的压强值，然后 FLUENT 就可以根据径向平衡关系计算出出口平面其余部分的压强值。径向平衡关系为：

$$\frac{\partial p}{\partial r}=\frac{\rho v_{\theta}^{2}}{r} \tag{8-26}$$

式中 r 为以转动轴为中心计算出的半径值，v_{θ} 是切向速度。径向平衡边界条件不仅可以用于有切向速度的入口边界，还可以用于切向速度为零的边界条件，比如用于计算带导向叶片的管道入口处的流动。另一个要注意的问题是，径向平衡入口条件只能用于三维轴对称旋转流计算。

2. 定义回流条件

回流条件也在压强出口条件中设定，但是仅在出口处发生回流现象时被使用。

（1）在包含能量计算的问题中需要设定回流总温（Backflow Total Temperature）。

（2）在回流的流动方向已知，并且与流场解相关时，则可以在 Backflow Direction Specification Method（回流方向定义方法）下拉列表中选择一种方法来定义回流方向。系统缺省设置是 Normal to Boundary（垂直于边界），即认为流动方向与边界平面垂直，这种情况下不需要另外输入其他参数。如果选择 Direction Vector（方向矢量）选项，则面板上会出现定义回流方向矢量分量的输入栏。如果计算使用的是三维求解器，则还会出现坐标系列表。如果选择 From Neighboring Cell（导自临近单元）选项，FLUENT 会使用紧邻压强出口的网格单元中的流动方向定义出口边界面上的流动方向。

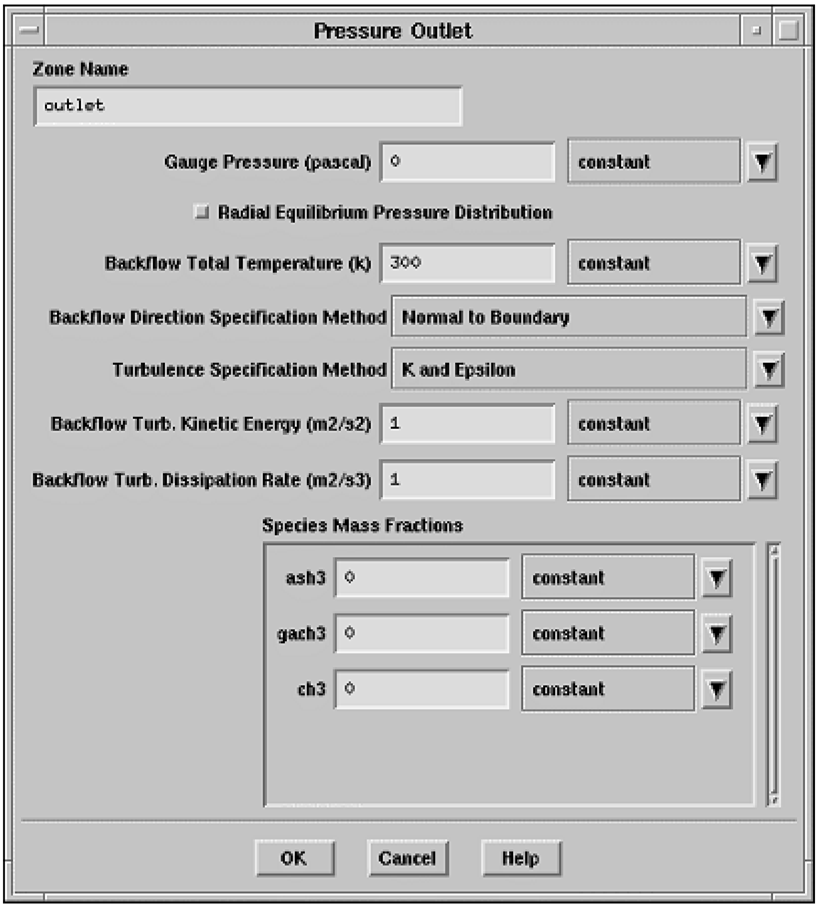


图 8-9 Pressure Outlet（压强出口）面板

（3）如果出现回流，Gauge Pressure（表压）一栏中的压强值将被作为总压使用，同时回流方向被认为是垂直于边界的。

如果紧邻压强出口边界的网格是移动的，并且求解器为分离求解器，则动压计算所采用的速度形式与 Solver（求解器）面板中选择的速度形式相同，即如果选择了绝对速度，动压就用绝对速度求出，如果选择的是相对速度，则动压用相对速度求出。对于耦合求解器，速度永远采用绝对速度形式。

即使在计算结果中没有回流出现，也应该将出口条件用真实流场的值设定，这样可以在计算过程中出现回流时加速收敛。湍流参数、辐射参数的设置可以参考相关章节的内容，这里不再重复。

8.8.2 压强出口边界的缺省设置

压强出口边界条件的缺省设置如下：

表 8-5 压强出口边界条件的缺省设置

参数	数值
表压（Gauge Pressure）	0
回流总温（Backflow Total Temperature）	300
回流湍流动能（Backflow Turb. Kinetic Energy）	1
回流湍流耗散率（Backflow Turb. Dissipation Rate）	1

8.8.3 压强出口边界的计算流程

在压强出口边界上，FLUENT 用 Gauge Pressure（表压）栏中的压强作为静压，其他变量则由流场内部插值获得。

可以用文本命令在压强出口边界上设置预期质量流量（Target Mass Flow Rate）。设置了这个选项后，在 FLUENT 的每个迭代步中都会调整压强的值，使得压强出口处的流量值满足目标流量的要求。调整流量的算法有两种：

1. 方法一

方法一由三步组成，为 FLUENT 设置的缺省计算方法，适用于涡轮机械计算。

第一步：计算压强出口边界所有网格面上静压的平均值，作为静压的初始值。

第二步：用压强出口边界上静压的平均值求出边界上的质量流量。用计算出的质量流量与目标流量的差额计算出压强增量，并对静压进行调整。将这一过程循环下去，使流量计算最终达到精度要求。

第三步：将第二步计算出的平均静压，经过亚松弛计算处理，设为边界条件。

2. 方法二

这种方法由两步组成，适用于带多重出口的复杂边界条件计算。

第一步：计算压强出口边界所有网格面上的平均静压和质量流量。

第二步：用目标质量流量、前面算出的质量流量和伯努利方程中的静压，计算出一个新的平均静压作为边界条件。计算中采用亚松弛方法保证计算的稳定性。

3. 启动目标质量流量选项

启动质量流量设置需要两个步骤，首先进入质量流量设置文本菜单（target – mass – flow – rate - settings text menu）：

define->boundary-conditions->target-mass-flow-rate-settings

回车后，在“>”号后面键入 enable 表示开启这个选项，然后键入字母“q”退出并返回上级菜单。键入 pressure-outlet 命令开始进行压强出口参数设置。在文本菜单出现预期质量流量选项时，键入 yes 表示开启这个选项。然后选择哪个压强出口需要使用目标流量设

置，以及目标流量的大小。

获得压强出口列表的方法如下：

define->boundary-conditions->pressure-outlet

键入相应的压强出口边界的名字，然后再为目标质量流量进行设置：

specify targeted mass-flow rate（给定目标质量流量）

输入 yes（是）或 no（否）

targeted mass-flow（目标质量流）

输入需要的值后，设置过程既告结束。

8.9 压强远场边界条件

压强远场条件用于设定无限远处的自由边界条件，主要设置项目为自由流马赫数和静参数条件。压强远场边界条件也被称为特征边界条件，因为这种边界条件使用特征变量定义边界上的流动变量。

采用压强远场边界条件要求密度用理想气体假设进行计算，为了满足“无限远”要求，计算边界需要距离物体相隔足够远的距离。比如在计算翼型绕流时，要求远场边界距离模型约 20 倍弦长左右。

8.9.1 压强远场边界条件的输入参数

在压强远场边界条件中需要输入下列参数：

（1）静压。

（2）马赫数。

（3）温度。

（4）流动方向。

（5）湍流计算中的湍流参数。

（6）辐射计算中的辐射参数。

（7）组元计算中的组元质量浓度。

（8）弥散相计算中的弥散相边界条件。

所有值均在压强远场（Pressure Far-Field）面板中输入，这个面板也是在 Boundary Conditions（边界条件）面板中启动。

其中的各种参数的设置与压强入口边界条件类似，故不再多做解释。

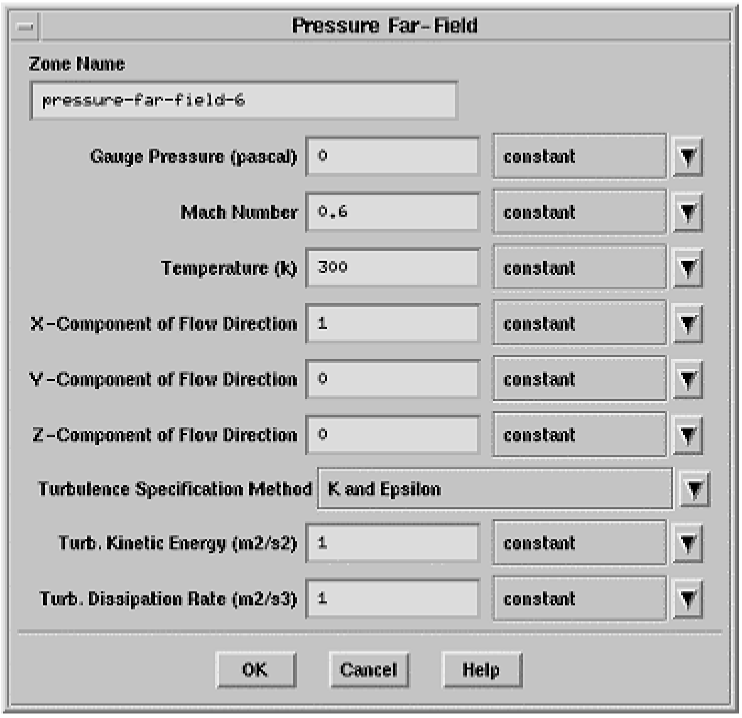


图 8-10 Pressure Far-Field（压强远场）面板

8.9.2 压强远场边界条件的缺省设置

压强远场边界条件的缺省设置如下：

表 8-6 压强远场边界条件的缺省设置

参数	数值
表压（Gauge Pressure）	0
马赫数（Mach Number）	0.6
温度（Temperature）	300
流动方向的 X 向分量（X-Component of Flow Direction）	1
流动方向的 Y 向分量（Y-Component of Flow Direction）	0
流动方向的 Z 向分量（Z-Component of Flow Direction）	0
湍流动能（Turb. Kinetic Energy）	1
湍流耗散率（Turb. Dissipation Rate）	1

8.9.3 压强远场边界条件的计算流程

压强远场边界条件是在局部一维假设下[2]通过引入黎曼不变量来定义的一类无反射边界条件。对于亚音速流动，存在两个黎曼不变量，分别对应于进入和离开流场的波。

$$R_\infty = v_{n\infty} - \frac{2c_\infty}{\gamma - 1} \tag{8-27}$$

$$R_i = v_{ni} + \frac{2c_i}{\gamma - 1} \tag{8-28}$$

式中v_n是法向速度，c是当地音速，γ是比热比。下标∞指远场边界，下标i指计算域内部（即紧邻边界的网格单元）。这两个不变量相加和相减后可以得到下面的方程：

$$v_n = \frac{1}{2}\left(R_i + R_\infty\right) \tag{8-29}$$

$$c = \frac{\gamma - 1}{4}\left(R_i - R_\infty\right) \tag{8-30}$$

这里v_n和c成为边界上的法向速度和音速。在流体流出的边界面上，切向速度分量和熵均由流场内变量插值得到。在流体流入的边界面上，这些量用自由来流的值定义。利用v_n、c、切向速度和熵，可以求出边界上的密度、速度、温度和压强。

8.10 出流边界条件

如果在流场求解前，流场出口处的流动速度和压强是未知的，就可以使用出流边界条件（outflow boundary conditions）。除非计算中包含辐射换热、弥散相等问题，在出流边界上不需要定义任何参数，FLUENT 用流场内部变量通过插值得到出流边界上的变量值。需要注意的是下列情况不适合采用出流边界条件：

（1）如果计算中使用了压强入口条件的话，应该同时使用压强出口条件。

（2）流场是可压流时。

（3）在非定常计算中，如果密度是变化的，则不适用出流边界条件。

[2] 局部一维流方向与边界面垂直。

8.10.1 FLUENT 对出流边界的处理方法

FLUENT 在出流边界上使用的边界条件如下：

（1）所有流动变量的扩散通量为零。

（2）总体质量平衡修正。

出流边界上的零扩散通量条件是指出流边界面上的流动变量均由流场内部变量值外插得出，并且对内部流场没有影响。FLUENT 在假定出流为充分发展流动，同时出口边界的面积不变的前提下，用外插计算在每次迭代中更新边界上的速度和压强。充分发展的流动是指流场变量在流动方向上不再发生变化[3]的流动。

8.10.2 出流边界条件的使用

如上节所述，出流边界条件服从充分发展流动假设，即所有流动变量的扩散通量在出口边界的法向等于零。在实际的计算中虽然不必拘泥于充分发展流动假设，但是只有在确信出口边界的流动与充分发展流动假设的偏离可以忽略不计时，才能使用出流边界条件。在出流边界存在很大的法向梯度，或者出现回流时不应使用出流边界条件。比如在分离点前后，流场速度梯度变化很大，并有可能出现回流，此时就不应该使用出流边界条件。

8.10.3 质量流分裂边界条件

在 FLUENT 软件中可以使用多个出流边界条件，并且定义每个边界上出流的比率。在 Outflow（出流）面板上，通过设置 Flow Rate Weighting（流量权重）就可以指定每个出流边界的流量比例。流量权重的计算公式如下：

$$\text{某个出流边界上流量百分比} = \frac{\text{设置在这个出流边界上的流量权重}}{\text{所有出流边界流量权重之和}} \qquad (8\text{-}31)$$

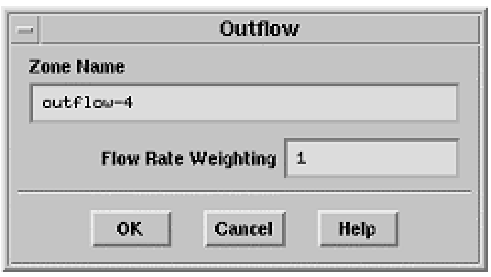


图 8-11 Outflow（出流）面板

[3] 流场变量在垂直于流动方向的截面上仍然允许变化。

在缺省设置中，所有出流边界的流量权重被设为 1。如果出流边界只有一个，或者流量在所有边界上是均匀分配的，则不必修改这项设置，系统会自动将流量权重的值进行调整，以使得流量在各个出口上均匀分布。比如有二个出流边界，而每个边界上流出的流量是总流量的一半，则无需修改缺省设置。但是如果有 75％的流量流出第一个边界，25％的流量流出第二个边界，则需要将第一个边界的流量权重修改为 0.75，第二个边界的流量权重修改为 0.25。

如果在计算中使用了辐射模型或弥散相模型，则需要针对这两个模型进行参数设置，具体方法请参见相关章节。

8.11 通风出口边界条件

出口通风条件用给定的损失系数和环境压强、温度定义出口边界条件。在这类边界条件中，出口处的通风设备被简化为一个边界面，并用边界上的压力损失代替通风损失。

在通风出口条件中需要输入下列参数：

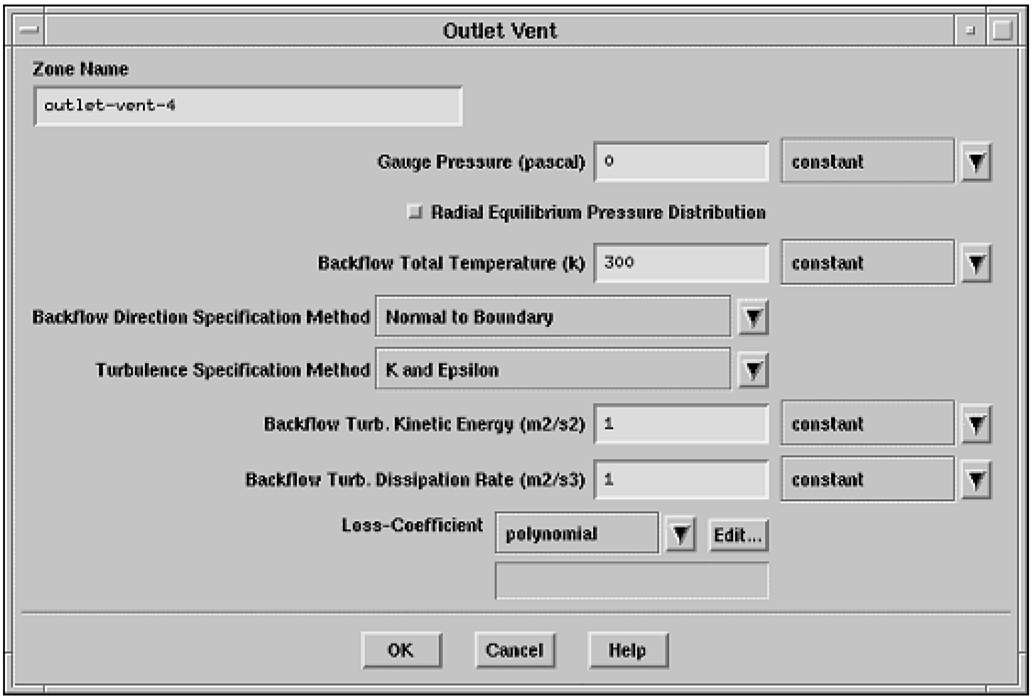


图 8-12 Outlet Vent（通风出口）面板

（1）静压。

（2）回流条件：

- 总温。

- 湍流参数。
- 化学组元质量浓度。
- 混合物浓度和增量。
- 燃烧模型中的过程变量。
- 多相流计算的多项流边界条件。

（3）辐射计算中的辐射参数。

（4）弥散相计算中的弥散相边界条件。

（5）损失系数。

这些参数均在 Outlet Vent（通风出口）面板中进行设置。其中前四项边界条件定义与压强入口边界条件一样，这里仅对损失系数的定义方法进行说明。

压力损失应该是总压损失，在考虑压力损失的时候，通常可以在伯努利方程中通过一个压力损失项来表征机械能的损失，而压力损失项则可以根据经验假定为与通风出口边界上的动压成正比，即：

$$\Delta p = k_L \frac{1}{2} \rho v_n^2 \qquad (8\text{-}32)$$

式中 Δp 为压力损失，k_L 就是通过经验确定的损失系数，ρ 是密度，v_n 是边界上的法向速度。

损失系数可以在 Loss Coefficient（损失系数）下拉列表中，用 constant（常数）、polynomial（多项式）、piecewise-linear（阶梯函数）或 piecewise-polynomial（分段多项式）等形式加以定义。

8.12 排气风扇边界条件

排气风扇边界条件用压强跃升和环境压强定义外部排气风扇。在排气风扇边界条件中，需要定义下列参数：

（1）静压。

（2）回流条件。

（3）辐射参数。

（4）弥散相边界条件。

（5）压强跃升。

这些参数均在 Exhaust Fan（排气风扇）面板中输入。在这些参数中前四项参数的定义方法在前面层做过介绍，这里仅介绍压强跃升的定义方法。

压强跃升可以在 Pressure Jump（压强跃升）下拉列表中，用 constant（常数）、polynomial（多项式）、piecewise-linear（阶梯函数）、piecewise-polynomial（分段多项式）等方式定义。

可以假定压强跃升是边界上法向速度的函数。排气风扇边界条件与回流条件配合使用，可以使得在流体流出边界时出现压强跃升，在回流时则相当于采用没有压力损失的入口通

风条件。

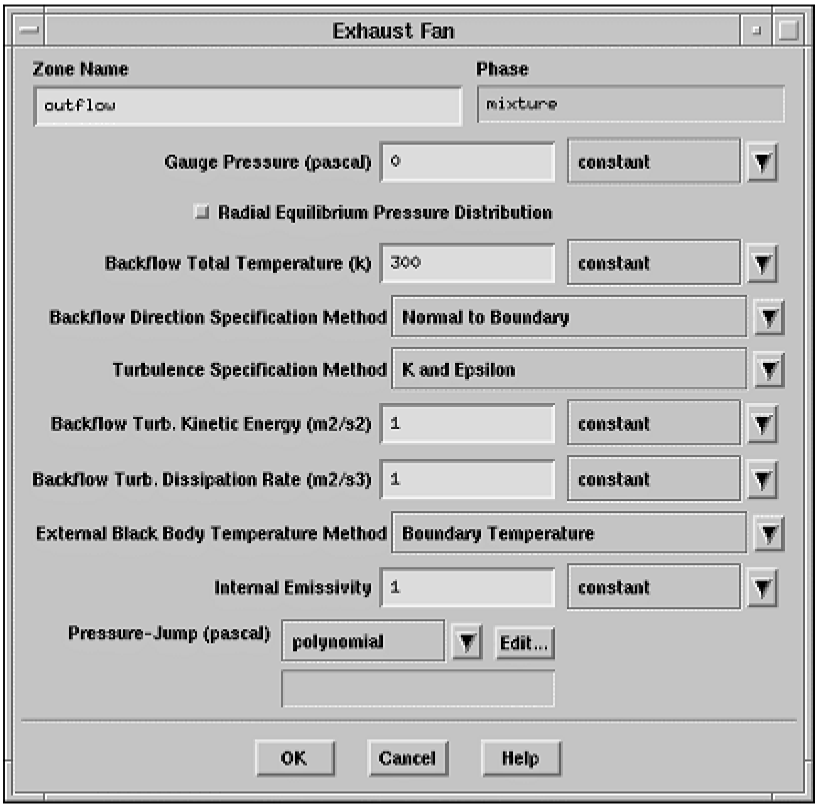


图 8-13 Exhaust Fan（排气风扇）面板

8.13 壁面边界条件

在粘性流计算中，FLUENT 使用无滑移条件作为缺省设置。在壁面有平移或转动时，也可以定义一个切向速度分量作为边界条件，或者定义剪切应力作为边界条件。

8.13.1 壁面边界条件中的输入参数

壁面边界条件上需要输入的参数如下：

（1）在热交换计算中的热力学边界条件。

（2）在移动、转动壁面计算中的壁面运动条件。

（3）滑移壁面中的剪切力条件。

（4）湍流计算中的壁面粗糙度。

（5）组元计算中的组元边界条件。

（6）在表面化学反应计算中的化学反应边界条件。
（7）辐射计算中的辐射边界条件。
（8）弥散相计算中的弥散相边界条件。
（9）VOF 计算中的多相流边界条件。

1. 定义热力学边界条件

在求解能量方程时，需要在壁面定义热力学边界条件。可以定义的热力学条件有 5 种：
（1）固定热通量。
（2）固定温度。
（3）对流热交换。
（4）外部辐射热交换。
（5）外部辐射与对流混合热交换。

如果壁面区域是所谓双侧壁面，即壁面为两个流动区域的交界面，则只能选择上述条件中的几项作为边界条件，但是同时却增加了确定正反两个侧面在计算中是否耦合这样的选项。如果壁面厚度不为零，还应该设置用于计算薄壁热阻和壁面生成热的参数。详情见下面的叙述：

首先，在计算中可以对普通壁面和内部壁面（即双侧壁面）定义热传导模型。这类热传导模型称为薄壳传导（shell conduction），可以更方便地用于壁面厚度与流场几何尺度相比很小时热传导问题的计算，比如带换热片的加热器周围流场的计算和汽车内部的薄金属板表面的热传导计算等等。如果在计算中将这类薄壁划分成固体区域网格的话，所生成的网格必然有很大的长细比，网格数量也将大大增加。

热力学条件在 Wall（壁面）面板的 Thermal（热力学）标签下输入，如图 8-14。下面逐项介绍相关参数的输入：

（1）热通量边界条件

在边界条件的热通量是固定值时，可以点击 Heat Flux（热通量）选项设置热通量。系统的缺省设置将热通量设为零，即假定壁面为绝热壁。计算中可以根据实际情况，在这一项中输入已知的热通量数据。

（2）温度边界条件

如果边界上的温度是固定值，则可以选择输入温度边界条件。只要点击选择 Temperature（温度）选项，然后在相应位置输入壁面温度值就完成了壁面边界条件的输入。

（3）对流热交换边界条件

选择 Convection（对流）选项，再输入 Heat Transfer Coefficient（热交换系数）和 Free Stream Temperature（自由流温度），FLUENT 就可以用后面的方程进行壁面上的热交换计算了。

（4）外部辐射边界条件

如果计算中需要考虑外界对流场的辐射，则应该选择设定 Radiation（辐射）条件，然后设定 External Emissivity（外部辐射率）和 External Radiation Temperature（外部辐射温度）。

（5）对流与外部辐射混合边界条件

选择 Mixed（混合）选项，可以同时设定对流与外部辐射边界条件。在这种情况下，可以设置的参数包括 Heat Transfer Coeffient（热交换系数）、Free Stream Temperature（自由流温度）、External Emissivity（外部辐射率）和 External Radiation Temperature（外部辐射温度）。

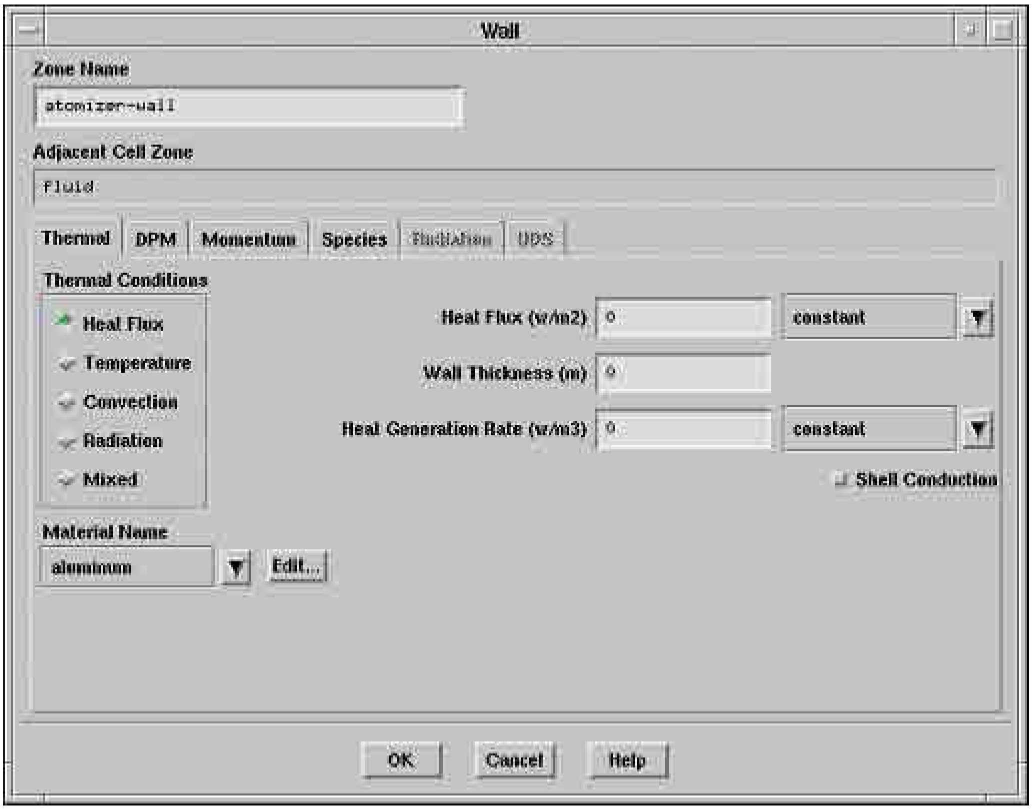


图 8-14　Wall（壁面）面板中的热力学分项

（6）薄壁热阻参数

在缺省设置中，壁面厚度等于零。但是在设定热力学条件的时候，可以在两个计算域之间定义一个带厚度的薄层。比如在计算插入流场中的一个薄金属板时，可以给予薄板一个厚度用于热力学计算。在这种情况下，FLUENT 在壁面附近用一维流假设计算由壁面引起的热阻和壁面上热量的生成量。

为了将这些效应引入计算，需要定义薄板的材料类型、厚度和热生成率。材料类型在 Material Name（材料名称）列表中选择，材料厚度在 Wall Thickness（壁面厚度）中设置。材料的物理属性可以在 Material（材料）面板中进行编辑。

壁面热阻为 $\Delta x / k$，其中 k 为壁面材料的热导率，Δx 为壁面厚度。热力学边界条件应该在薄壁的外侧进行设置，如图 8-15 所示，图中 T_b 是壁面上设置的固定温度。

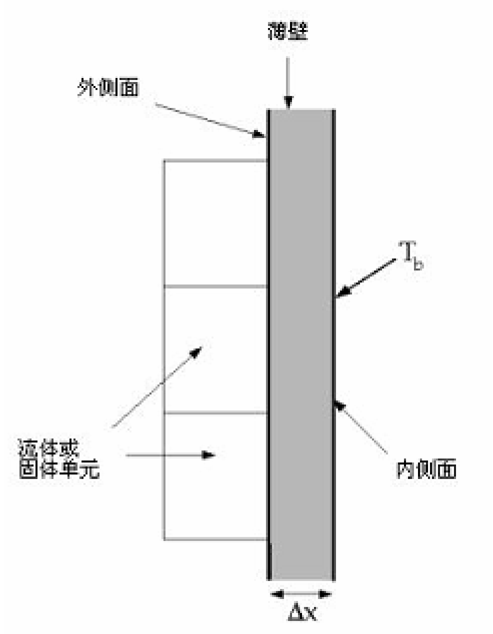


图 8-15 薄壁热力学边界条件

在薄壁条件中只能定义一个常数的热导率。如果要对一个非零厚度的壁面定义可变的热导率，则需要使用薄壳热导模型（shell conduction model）。

在 Heat Generation Rate（热生成率）中定义壁面的热生成率。这个选项是针对已知壁面热生成率的情况设置的，例如在计算印刷电路板问题时，电路板上的热生成率就可以从电功消耗中计算得出，因而适用于设定热生成率。

（7）双侧壁面的热力学边界条件

如果壁面两侧均为计算域，则称为双侧壁面。这种类型的网格文件读入 FLUENT 后，FLUENT 中将自动生成影子（shadow）区域，即壁面的每个面都有一个计算区域与之一一对应。在 Wall（壁面）面板中，影子区域的名字显示在 Shadow Face Zone（影子表面区域）中。对互为影子的两个面可以分别定义不同的热力学条件，也可以将二者耦合起来进行定义：

- 如果想将二者耦合，可以在 Thermal Conditions（热力学条件）中选择 Coupled（耦合）选项，这个选项只在壁面为双侧壁面时出现。无需设定其他的热力学参数，求解器可以直接用邻近网格的流场变量计算壁面上的热交换，但是计算中涉及到的材料类型、壁面厚度和用于薄壁热阻计算的热生成率都可以用前面讲到的方法输入。在一侧壁面设定的热阻参数将自动被用于它的影子壁面，这就是耦合的特点。
- 如果不想将薄壁的两个面耦合在一起进行计算，也可以通过选择 Thermal Conditions（热力学条件）下的 Temperture（温度）或 Heat Flux（热通量）选项，分别设定两个面的热力学条件。需要注意的是，对于双侧壁面无法设置对流和辐射条件。壁面与它的影子之间的关系将被保存，所以在需要的时候，还可以将边

界条件重新设定为耦合关系。在双侧壁面被解耦之后，两个壁面可以有不同的壁面厚度，并且相互之间是绝热的。如果为解耦的壁面设定了非零厚度，那么热力学条件将被定义在壁面的外侧，如图 8-16 所示。

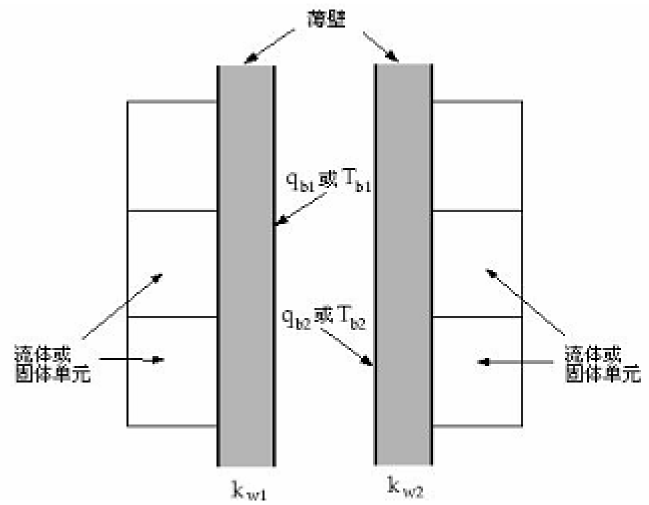


图 8-16 解耦后壁面上的热力学条件

图中T_{b1}和q_{b1}分别是设定在一个壁面上的温度和热通量；T_{b2}和q_{b2}则分别是另一个壁面上的温度和热通量。k_{w1}和k_{w2}是解耦后薄壁上的热导率。注意图中两个壁面之间的空隙并不是模型的一部分，图上这样画仅仅是为了表示热力学边界条件是定义在壁面的哪一侧。

（8）壁面上的薄壳热导率

在壁面边界条件中选中薄壳热导率（Shell Conduction）就可以用定义薄壳热导率的形式定义热力学边界条件。在使用这种方式定义热力学边界条件的时候，热力学条件的定义方法与前面的薄壁条件定义方法相同。

需要注意的是，必须定义一个非零壁面厚度，因为薄壳热导率计算需要一个非零厚度的值。另外，采用薄壳条件有下面几项限制：

- 薄壳热导率定义仅适用于三维计算。
- 仅在使用分离算法求解器时使用。
- 在使用非预混和部分预混燃烧模型时不能使用。
- 在使用多相流混合物模型、VOF 模型或欧拉(Eulerian)模型时不能使用。
- 在与 DO 辐射模型同时使用时，薄壳壁面不能被设为半透（semi-transparent）壁面。
- 薄壳壁面不能被分割或合并。如果需要分割或合并一个薄壳壁面，首先需要取消薄壳壁面设置，然后进行分割或合并操作，再重新对新生成的壁面设定薄壳选项。
- 薄壳热导率模型不能用于做过适应性调整的壁面区域。
- 薄壳壁面在端部的热通量不会在热平衡报表中出现。这些通量仅用于 FLUENT 的

求解过程，并不出现在报告中。

2. 定义壁面运动

壁面边界可以是静止的，也可以是运动的。移动壁面边界条件采用壁面的平移或转动的速度或速度分量值加以定义。

壁面运动是在 Wall（壁面）面板的 Momentum（动量）部分进行定义的，点击 Momentum（动量）标签可以看到与壁面运动有关的所有定义形式。

（1）定义静止壁面

在 Wall Motion（壁面运动）下选择 Stationary Wall（静止壁面），就可以将壁面设置为静止壁面。

（2）定义运动壁面的速度

如果计算中壁面存在切向运动，就需要在边界条件中定义平移或转动速度，或速度分量。在 Wall Motion（壁面运动）下选择 Moving Wall（移动壁面），则 Wall（壁面）面板随即展开如图 8-17 所示。注意在移动壁面条件中不能设定壁面的法向运动，FLUENT 会忽略所有法向移动速度。

（3）定义相对或绝对速度

如果壁面附近的网格是移动网格，则可以选择用相对速度的方式定义壁面运动，即取移动网格为参考系定义壁面的运动速度。此时只要点击选择“Relative to Adjacent Cell Zone（相对于临近网格）”选项就可以了。

如果选择了 Absolute（绝对速度）选项，则可以通过定义壁面在绝对坐标系中的速度来定义壁面运动。如果临近的网格单元是静止的，则相对速度和绝对速度的定义是等价的。

（4）壁面的平移运动

在壁面存在直线平移运动时，可以选择 Translational（平移）选项，并在 Speed（速度）和 Direction（方向）栏中定义壁面运动速度矢量。缺省情况下，系统认为壁面是静止的，所以速度值被设为零。

（5）壁面的旋转运动

选择旋转（Rotational）选项并确定绕指定转动轴的旋转速度，就可以将壁面的旋转运动唯一确定下来。用 Rotation-Axis Direction（转动轴方向）和 Rotation-Axis Origin（转动轴原点）可以唯一确定转动轴。在三维计算中，转动轴是通过转动轴原点并平行于转动轴方向的直线。在二维计算中，无需指定转动轴方向，只需指定转动轴原点，转动轴是通过原点并与 z 方向平行的直线。在二维轴对称问题中，转动轴永远是 x 轴。

（6）用速度分量定义壁面运动

选择 Components（速度分量）选项，就可以通过定义壁面运动的速度分量来定义壁面的平移运动。这里定义的平移运动可以是直线运动，也可以是非直线运动。其运动方式可以用速度分量函数或自定义函数的形式加以定义。

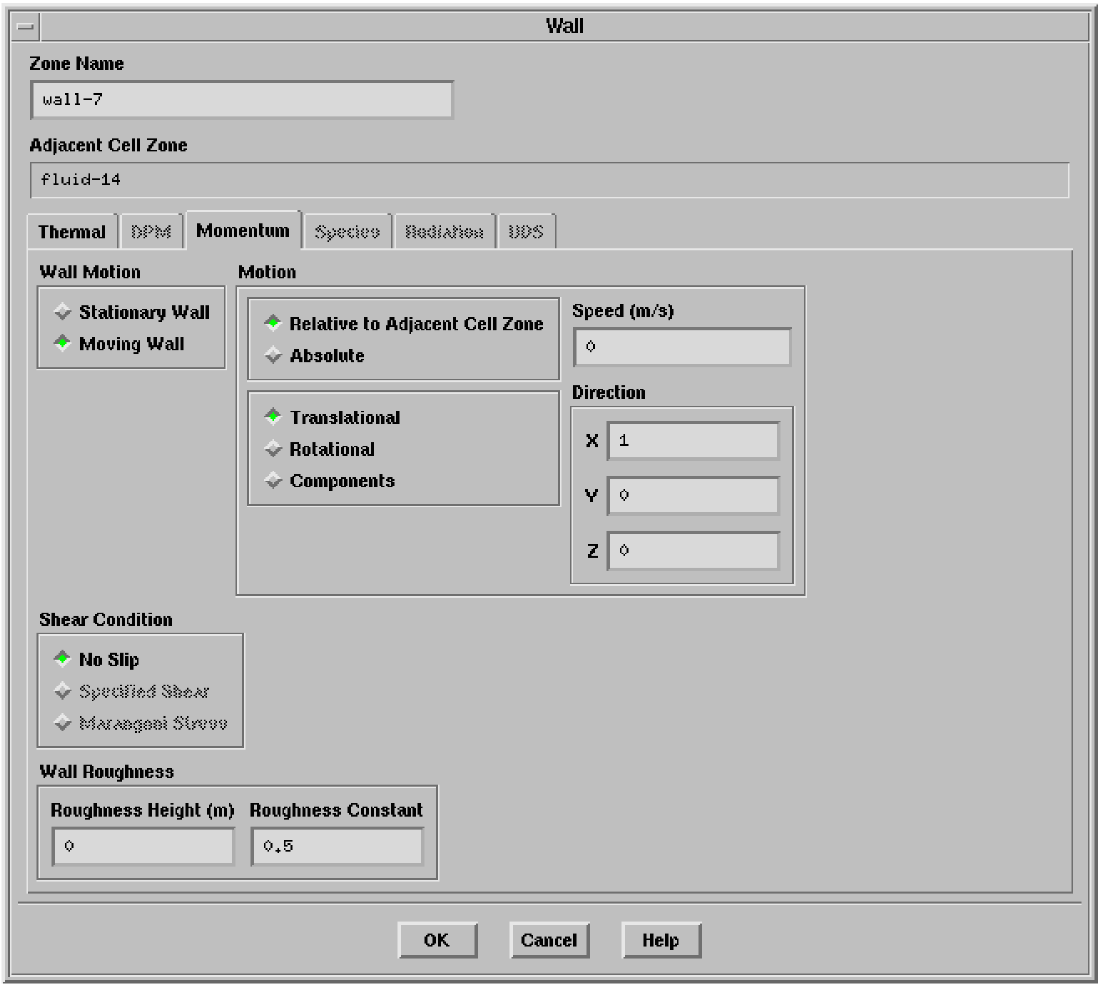


图 8-17 移动壁面（Moving Wall）定义

（7）双侧壁面的运动

对于双侧壁面，可以对壁面及其影子区域定义不同的运动，无论它们耦合与否。如果与壁面相邻的区域是固体区域，则无法定义其运动。

3. 定义壁面上的剪切条件

可以定义三种类型的剪切条件，即：

（1）无滑移条件。

（2）指定剪切力条件。

（3）Marangoni 应力条件。

对于所有移动壁面只能设定无滑移条件，其它类型的剪切条件仅适用于静止壁面。

无滑移条件是系统缺省设置，其物理含义是紧邻壁面的流体将与壁面结合在一起，并以相同速度运动的意思。

指定剪切力条件和 Marangoni 应力是在剪切力已知的时候使用的。指定剪切力的边界条件允许定义剪切力的 x、y、z 分量等于常数，或用边界上的型函数定义。Marangoni 应力边界条件允许根据表面温度定义表面应力梯度。剪切应力是用温度在表面附近的梯度值和表面张力梯度计算得出的。Marangoni 应力只在求解过程包含能量方程的求解时能够使用。

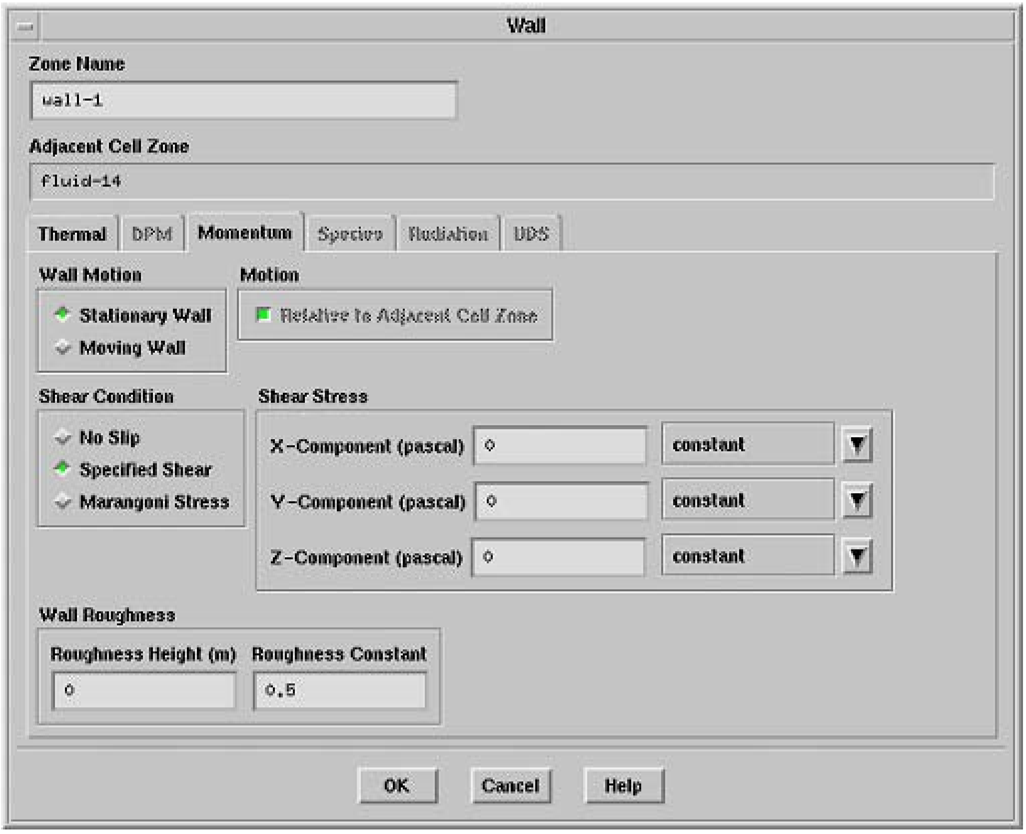


图 8-18 指定剪切力条件的定义

剪切力条件在壁面（Wall）面板中的动量（Momentum）部分输入。

（1）定义无滑移条件

在 Shear Condition（剪切条件）下选择 No Slip（无滑移）选项就可以在壁面上设定无滑移条件。无滑移条件是粘性流计算中所有壁面的缺省设置。

（2）指定剪切力条件

在剪切条件下选择 Specified Shear（指定剪切力）选项就可以为壁面设定剪切力的值，如图 8-18。然后可以通过输入剪切力的 x、y、z 分量来定义剪切力的值。在剪切力给定后，湍流计算中的壁面函数条件就不再使用。

（3）Marangoni 应力

FLUENT 可以定义由温度引起的表面张力的变化。壁面上的剪切张力定义如下：

$$\tau = \frac{d\sigma}{dT}\nabla_s T \qquad (8\text{-}31)$$

式中 $d\sigma / dT$ 是表面张力对温度的梯度，$\nabla_s T$ 是温度在表面上的梯度。上式定义的剪切力直接用于动量方程计算。

在 Shear Condition（剪切力条件）下选择 Marangoni Stress（Marangoni 应力）选项，就可以在壁面上定义 Marangoni 应力条件，如图 8-19。在表面张力梯度中可以输入表面张力梯度的值。

4. 壁面粗糙度对壁面限定湍流的影响

壁面粗糙度对流动阻力和传热、传质都有影响，在湍流计算中可以通过加入粗糙度影响的方式对壁面律做出修正。

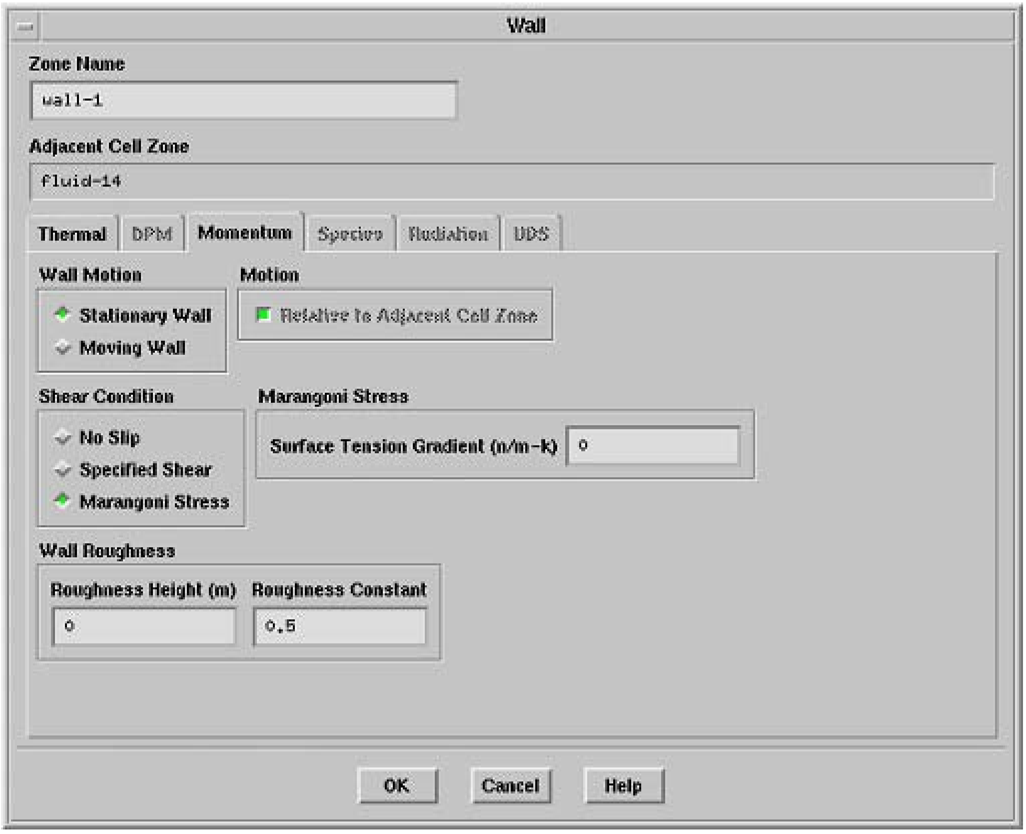


图 8-19 Marangoni 应力条件设定

（1）加入粗糙度修正的壁面律

在粗糙管道和槽道中的试验结果标明，靠近粗糙壁面的平均流速分布在引入粗糙度修正后满足下式定义的规律：

$$\frac{u_p u^*}{\tau_w/\rho}=\frac{1}{k}\ln\left(E\frac{\rho u^* y_p}{\mu}\right)-\Delta B \qquad (8\text{-}33)$$

式中$u^*=C_\mu^{1/4}k^{1/2}$，$\Delta B=\frac{1}{k}\ln f_r$，$f_r$为粗糙度函数。$\Delta B$取决于粗糙度的类型（均匀的沙子、铆钉、螺纹、肋板、网线等等）和尺寸。不存在统一的适用于任意种类粗糙度的函数。对于均匀的沙子之类的“均匀”的粗糙颗粒，ΔB与无量纲的粗糙高度$K_s^+=\rho K_s u^*/\mu$相关，式中的K_s是粗糙颗粒的真实高度，同时$u^*=C_\mu^{1/4}k^{1/2}$。对试验数据的分析表明粗糙度函数是K_s^+的分段函数，即在K_s^+的不同取值范围上有不同的形式，K_s^+的取值范围可以分为三个：

1）水力光滑区（K_s^+≤2.25）

2）过渡区（2.25<K_s^+≤90）

3）完全粗糙区（K_s^+>90）

根据试验结果，在水力光滑区粗糙度的影响可以忽略不计，在过渡区逐渐发挥作用，而在完全粗糙区则起主导作用。FLUENT 中用于计算三个区中粗糙度影响的公式为：

1）水力光滑区：

$$\Delta B=0 \qquad (8\text{-}34)$$

2）过渡区：

$$\Delta B=\frac{1}{k}\ln\left[\frac{K_s^+-2.25}{87.75}+C_s K_s^+\right]\cdot\sin\left[0.4258\left(\ln K_s^+-0.811\right)\right] \qquad (8\text{-}35)$$

式中C_s是粗糙度常数，具体取值取决于粗糙颗粒的类型。

3）完全粗糙区：

$$\Delta B=\frac{1}{k}\ln\left(1+C_s K_s^+\right) \qquad (8\text{-}36)$$

在求解过程中，用上述三个公式计算出ΔB的值，然后再将ΔB代入（8-32），并根据湍流变量和平均温度求出壁面剪切力。

（2）设置粗糙度参数

粗糙度参数是在 Wall（壁面）面板中的 Momentum（动量）部分中设定，需要输入的参数有两个，即 Roughness Height（粗糙颗粒高度），K_s，以及 Roughness Constant（粗糙

常数），C_s。系统缺省的粗糙高度 K_s 等于零，物理上对应于理想光滑管道。对于均匀沙粒型粗糙颗粒，K_s 就是沙粒的高度。对于非均匀沙粒型粗糙颗粒，K_s 可以取为颗粒的平均直径。对于其他类型的粗糙颗粒，则可以使用“等价”的颗粒高度作为 K_s 的值。

粗糙度常数 C_s 主要取决于粗糙颗粒的类型，系统中的缺省值 $C_s = 0.5$，在与 $k-\varepsilon$ 模型混合使用时，C_s 的缺省值可以准确地计算均匀沙粒型粗糙度。在粗糙颗粒的类型与均匀沙粒相距甚远，以致计算结果出现很大偏差时，可以调整 C_s 的取值。比如对于非均匀沙粒、肋板等粗糙度，可以在 0.5 到 1.0 之间进行选择。不过对于任意类型的粗糙颗粒还不存在一个通用的参数选择准则。

需要注意的一点是，没有必要让壁面附近的网格尺度小于粗糙度高度。为了获得最好的计算结果，需要保证从壁面到网格几何中心的尺寸大于粗糙度高度 K_s。

5. 定义壁面上的组元边界条件

在缺省设置中，除了参与壁面反应的组元，所有组元在壁面附近的梯度为零，但是同时也可以设定壁面上的质量浓度。就是说，在入口边界上采用的 Dirichlet 边界条件也可以用于壁面边界。

如果系统的缺省设置不能满足要求，可以用下列步骤进行修改：

（1）点击 Wall（壁面）面板的 Species（组元）标签，可以看到壁面上的组元边界条件。

（2）在 Species Boundary Condition（组元边界条件）下面，组元名称右面的下拉列表中选择 Specified Mass Fraction（指定质量浓度），而不是缺省的 Zero Diffusive Flux（零扩散通量），则面板会展开出现定义 Species Mass Fractions（组元质量浓度）选项，如图 8-20。

（3）在组元质量浓度中定义组元的质量浓度。

每个组元的边界条件都是单独定义的，所以在定义组元边界条件时可以对不同组元采用不同的定义方法。

如果某个组元参与了壁面反应，也可以点击 Reaction（反应）选项并从 Reaction Mechanisms（反应机制）列表中选择相应的反应，为组元设定反应。

6. 定义壁面上的反应边界条件

在图 8-18 中开启或关闭表面反应（Surface Reactions）选项，就可以设定壁面条件中是否包含表面反应。

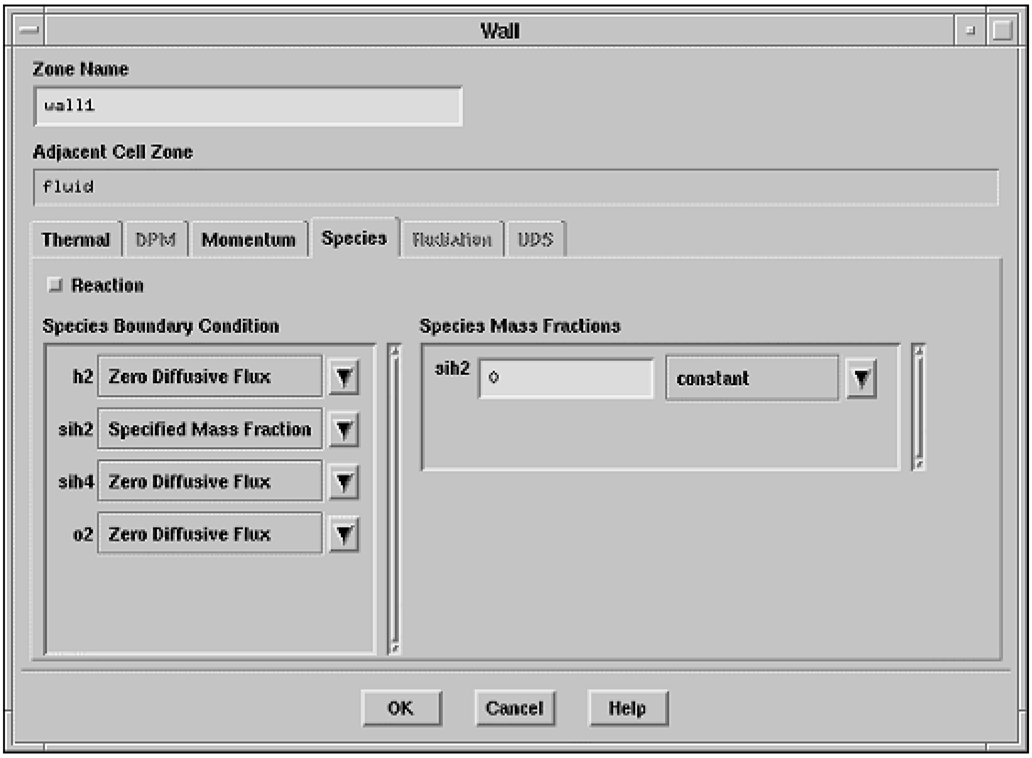


图 8-20 Species（组元）边界条件

7. 定义壁面上的辐射边界条件

如果计算中使用了 P-1 模型、DTRM 模型、DO 模型或面到面模型，则需要在 Wall（壁面）面板的 Radiation（辐射）部分设定壁面辐射率。如果使用的是 Rosseland 模型，则无需设定任何参数，因为 FLUENT 已经将辐射率设定为 1。如果使用的是 DO 模型，还需要设定壁面类型为扩散型、镜面型或半透型。

8. 定义壁面上的弥散相边界条件

如果计算中使用了弥散相模型，则需要在 Wall（壁面）面板的 DPM 部分设定粒子轨迹的限定条件。

9. 定义壁面上的多相流边界条件

如果计算中使用了 VOF 模型，则可以在 Wall（壁面）面板的 Momentum（动量）部分定义两相之间的接触角。

10. 定义用户自定义变量的边界条件

如果计算中用到了用户自定义参数，则可以在 Wall（壁面）面板的 UDS 部分设置其边

界条件。

8.13.2 壁面边界条件的缺省设置

壁面上的缺省设置是热通量等于零，同时壁面是静止不动的。

8.13.3 壁面边界条件中剪切应力的计算方法

在无滑移条件下，FLUENT 使用临近网格中的流体属性计算壁面上的剪切应力。在层流计算时，计算只与壁面上的速度梯度有关，而在湍流计算中则需要使用壁面函数。对于给定剪切力的壁面，FLUENT 将在边界上计算切向速度。

在无粘流计算中，所有壁面均使用滑移条件，因此壁面为无摩擦壁面，对临近流体没有剪切应力。

1. 层流中的剪切应力计算

层流中壁面剪切应力用法向速度梯度计算：

$$\tau_w = \mu \frac{\partial v}{\partial n} \qquad (8\text{-}37)$$

当壁面附近存在较大速度梯度时，必须采用密集网格以保证计算可以正确捕捉边界层。

2. 湍流中的剪切应力计算

在湍流相关章节中将详细讨论这个问题，主要的思路是在壁面附近采用壁面函数等特殊的处理方法。

8.13.4 壁面边界上的热传导计算

1. 温度边界条件

如果在壁面上设定了固定温度作为边界条件，则壁面上的热通量计算公式为：

$$q = h_f\left(T_w - T_f\right) + q_{rad} \qquad (8\text{-}38)$$

式中 h_f 为流体的热传导系数，T_w 为壁面温度，T_f 为当地流体温度，q_{rad} 为辐射热通量。流体的热传导系数 h_f 是用当地流场条件（湍流度、温度和速度型函数）求出的。

由固体壁面产生的热交换计算公式为：

$$q = \frac{k_s}{\Delta n}\left(T_w - T_s\right) + q_{rad} \qquad (8\text{-}39)$$

式中 k_s 为固体的热导率，T_s 为固体温度，Δn 为壁面与固体单元几何中心的距离。

2. 热通量边界条件

在选择定义壁面热流通量时，FLUENT 用方程（8-37）计算热流通量，并用下式计算壁面温度：

$$T_w = \frac{q - q_{rad}}{h_f} + T_f \qquad (8\text{-}40)$$

如果壁面包围的是固体区域，则壁面温度用下式计算：

$$T_w = \frac{\left(q - q_{rad}\right)\Delta n}{k_s} + T_s \qquad (8\text{-}41)$$

3. 对流热交换边界条件

在采用对流热交换边界条件时，FLUENT 用下面的公式计算壁面热流通量：

$$q = h_f\left(T_w - T_f\right) + q_{rad} = h_{ext}\left(T_{ext} - T_w\right) \qquad (8\text{-}42)$$

式中 h_{ext} 为用户定义的外部热交换系数，T_{ext} 为用户定义的外部热源温度，q_{rad} 为辐射热通量。

方程（8-41）假定壁面为零厚度壁。

4. 外部辐射边界条件

在考虑外部辐射时，FLUENT 用下面的公式计算热流通量：

$$q = h_f\left(T_w - T_f\right) + q_{rad} = \varepsilon_{ext}\sigma\left(T_\infty^4 - T_w^4\right) \qquad (8\text{-}43)$$

式中 ε_{ext} 为用户定义的外部壁面辐射率，σ 为 Stefan-Boltzmann 常数，T_w 为壁面的表面温度，T_∞ 为用户定义的热源温度，q_{rad} 从计算域到边界的辐射热流通量。上述方程同样假定壁面为零厚度壁。

5. 外部对流和辐射的混合边界条件

在使用混合热交换条件时，壁面热流通量可以用下式计算：

$$q = h_f\left(T_w - T_f\right) + q_{rad} = h_{ext}\left(T_{ext} - T_w\right) + \varepsilon_{ext}\sigma\left(T_\infty^4 - T_w^4\right) \qquad (8\text{-}44)$$

式中各参数与前面公式中的含义相同。这个公式也假定壁面为零厚度壁面。

6. 流体热交换系数的计算

在层流计算中，壁面上的热交换用 Fourier 级数展开形式进行计算。FLUENT 中采用其离散形式进行计算，即：

$$q = k_f\left(\frac{\partial T}{\partial n}\right)_{wall} \qquad (8\text{-}45)$$

式中 n 为当地法向坐标。

在湍流计算中，FLUENT 利用热交换和动量交换的相似性得到温度的壁面函数形式，详情请参考本书湍流计算部分的内容。

8.14 对称边界条件

在流场内的流动及边界形状具有镜像对称性时，可以在计算中设定使用对称边界条件。这种条件也可以用来定义粘性流动中的零剪切力滑移壁面。本节将讲述在对称面上对流体的处理方式，并给出相关的实例。在对称边界上不需要设定任何边界条件，但是必须正确定义对称边界的位置。

需要注意的是，在轴对称流场的对称轴上应该使用轴（Axis）边界条件，而不是对称边界条件。

8.14.1 对称边界的例子

采用对称边界条件可以将计算域大大缩小，进而大大降低计算工作量，加快计算速度。图 8-19 是一个对称边界的实例。

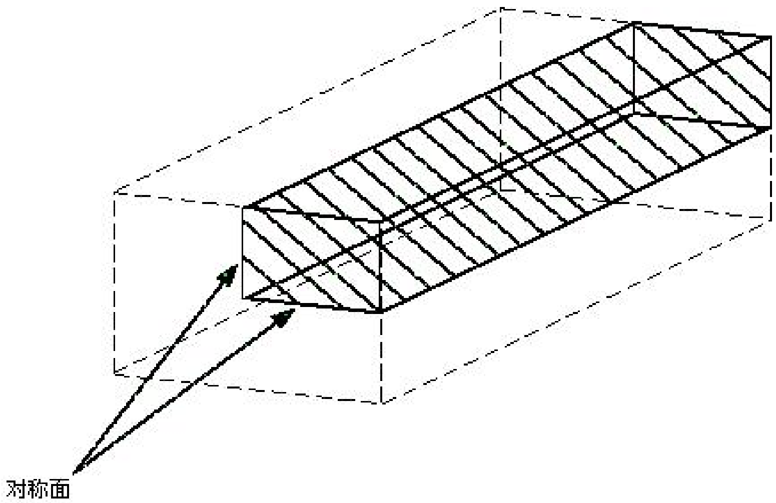
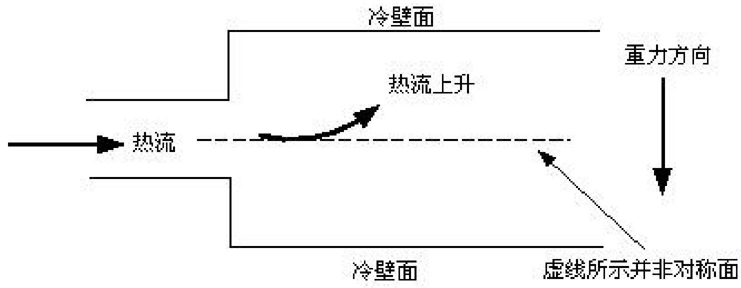


图 8-19 对称面的例子

图 8-20 是两个不适用对称边界的实例。这两个例子中，虽然边界是几何对称的，但是流场结构并不对称。在第一个例子中，由于浮力作用产生了非对称流动。在第二个例子中，由于流动带有旋转，所以使得整个流场结构不存在“镜像”对称条件，而适合采用后面提到的旋转周期边界。



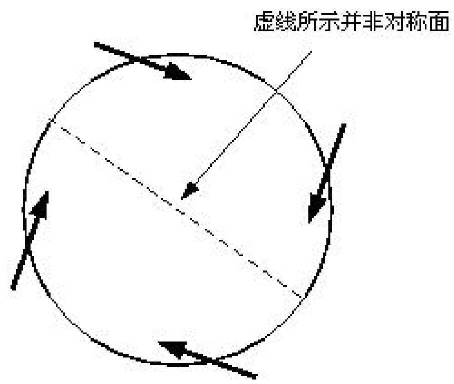


图 8-20 不适用对称边界条件的例子

8.14.2 对称边界上的计算流程

在对称面上所有流动变量的通量为零。由于对称面上的法向速度为零，所以通过对称面的对流通量等于零。对称面上也不存在扩散通量，因此所有流动变量在对称面上的法向梯度也等于零。对称边界条件可以总结为：

（1）对称面上法向速度为零。

（2）对称面上所有变量的法向梯度为零。

如上所述，对称面的含义就是零通量。因为对称面上剪切应力等于零，在粘性计算中对称面条件也可以被称为“滑移”壁面。

8.15 周期性边界条件

在流场的边界形状和流场结构存在周期性变化特征时，可以采用周期性边界条件。在 FLUENT 中可以设置两种周期性边界条件。第一种类型的周期性边界条件不允许在周期平

面上出现压力降。第二种周期性边界条件则允许在周期边界上出现压强降，从而可以计算“充分发展”的周期性流动。

本小节主要讨论第一种周期性边界条件，第二种则在讲述周期性流动的章节中专门进行讨论。

8.15.1 周期性边界条件的例子

当流场中两个相对平面上的流动完全相同时，可以采用周期性边界条件。图 8-21 是周期性边界条件的一个典型例子。在这个例子中，流入计算域的平面和流出计算域的平面上的流动是完全一样的。就象例子中表现的那样，周期性边界总是成对出现的。

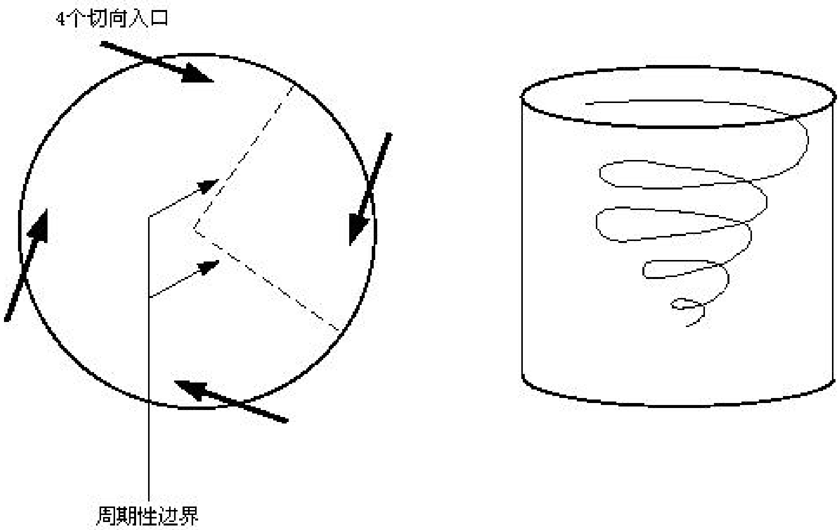


图 8-21 用周期性边界定义圆柱形容器中的旋转流动

8.15.2 周期性边界条件的输入

在不考虑压力降时，周期性边界中只需要输入一个参数，即定义几何边界是旋转周期性边界还是平移周期性边界的参数。

旋转周期性边界是绕一个中心线转过一定的角度后出现的周期性边界，平移周期性边界则是平移过流场时出现的周期性边界。图 8-21 中的周期性边界是旋转式的，图 8-22 中的则是平移式的。

在 Periodic（周期性边界）面板中设定边界为旋转式的或平移式的。

如果计算域是旋转式周期性边界，则在 Periodic Type（周期类型）中选择 Rotational（旋转），如果是平移式周期性边界，则选择 Translational（平移）。对于旋转式周期域，求解器将自动计算周期边界的转角，转动轴就取为临近网格单元的转轴。

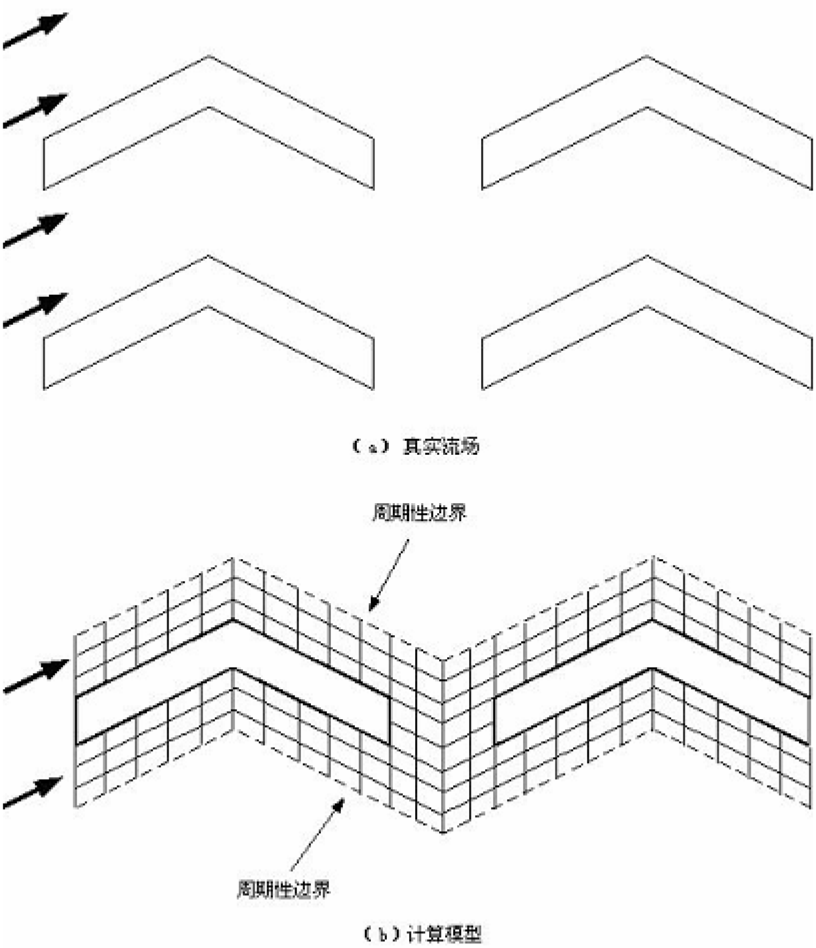


图 8-22 平移式周期性边界

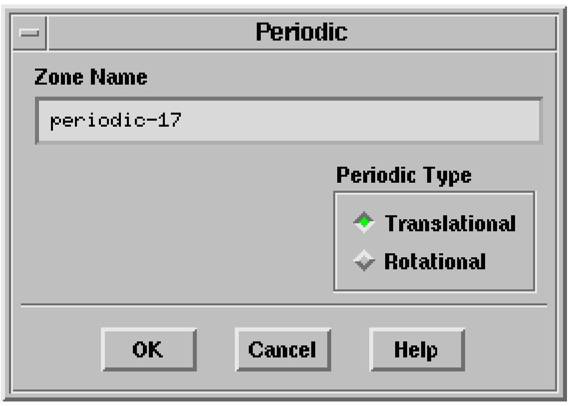


图 8-23 周期性边界面板

旋转式周期性边界的一个例子是三维管道流动。在计算三维管道流动时，可以不必计

算整个管道的流动，而只要在管道截面上用半径切出一个角就可以了，这个角的两个边上可以设定周期性边界条件。

可以用 Grid/Check 菜单项计算并显示周期性边界上的最小、最大和平均转动角度。如果这三者的差值很大，则说明网格划分有问题——网格在几何上不具备周期性边界特征。

在系统的缺省设定中，周期性边界是平移式周期边界。

8.15.3 周期性边界上的计算流程

周期性边界是成对出现的，FLUENT 将出口处的周期边界看作入口处周期边界的相邻边界，因此在计算周期边界的流动通量时，可以将入口边界内侧网格单元中的流场变量看作出口边界外侧的流场变量。从这个假定出发，在计算边界上的通量时，可以用入口内侧变量与出口内侧变量作为周期边界（入口和出口）两侧的变量算出周期边界上的流动通量。

8.16 轴边界条件

轴边界类型必须是轴对称几何边界的中心线，或者是四分之一圆柱、八分之一圆柱区域的转动轴。在轴边界上不用定义边界条件。轴上的流动变量值由轴的临近单元中的流场变量值计算得出。

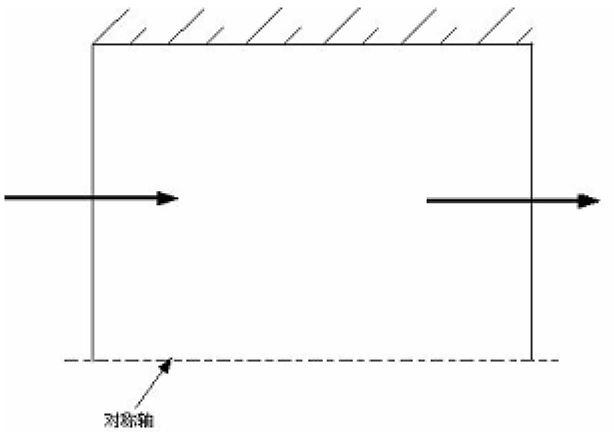


图 8-24 用轴对称几何的中心线做轴边界

8.17 流体条件

流体区域是网格单元的集合，所有需要求解的方程都要在流体区域上被求解。流体区

域上需要输入的唯一信息是流体的材料性质，即在计算之前必须指定流体区域中包含何种流体。

在计算组元输运或燃烧问题时不需要选择材料，因为在组元计算中流体是由多种组元组成的，而组元的特性在 Species Model（组元模型）面板中输入。同样在多相流计算中也不需要指定材料性质，流体的属性在指定相特征时被确定。

其他可以选择输入的参数包括源项、流体质量、动量、热或温度、湍流、组元等流动变量。还可以定义流体区域的运动。如果流体区域附近存在旋转式周期性边界，则需要指定转动轴。如果计算中使用了湍流模型，还可以将流体区域定义为层流区。如果计算中使用 DO 模型计算辐射，还可以确定流体是否参与辐射过程。

8.17.1 流体区域参数的输入

与流体区域相关的参数应该在 Fluid（流体）面板中输入，这个面板也是从 Boundary Conditions（边界条件）面板中启动。

1. 定义流体属性

可以从材料列表中选择材料，如果材料参数不符合要求，还可以编辑材料参数以便满足计算要求。

2. 定义源项

在 Source Terms（源项）选项中可以定义热、质量、动量、湍流、组元和其他流动变量的源项。

3. 定义固定参数值

选中 Fixed Values（固定值）选项可以为流体区域中的变量设置固定值。

4. 设定层流区

在计算中使用了 $k-\varepsilon$ 模型、$k-\omega$ 模型或 Spalart-Allmaras 模型时，可以在特定的区间上关闭湍流设置，从而设定一个层流区域。这个功能在已知转捩点位置，或层流区和湍流区位置时是非常有用的。

5. 定义化学反应机制

选中 Reaction（反应）选项后可以在 Reaction Mechanisms（反应机制）列表中选择需要的反应机制，从而可以在计算中计算带化学反应的组元输运过程。

6. 定义旋转轴

如果流体区域周围存在周期性边界，或者流体区域是旋转的，则必须为计算指定转动轴。通过定义 Rotation-Axis Direction（旋转轴方向）和 Rotation-Axis Origin（转轴原点）即可以定义三维问题中的转动轴。在二维问题中则只需要指定转轴原点就可以确定转动轴。

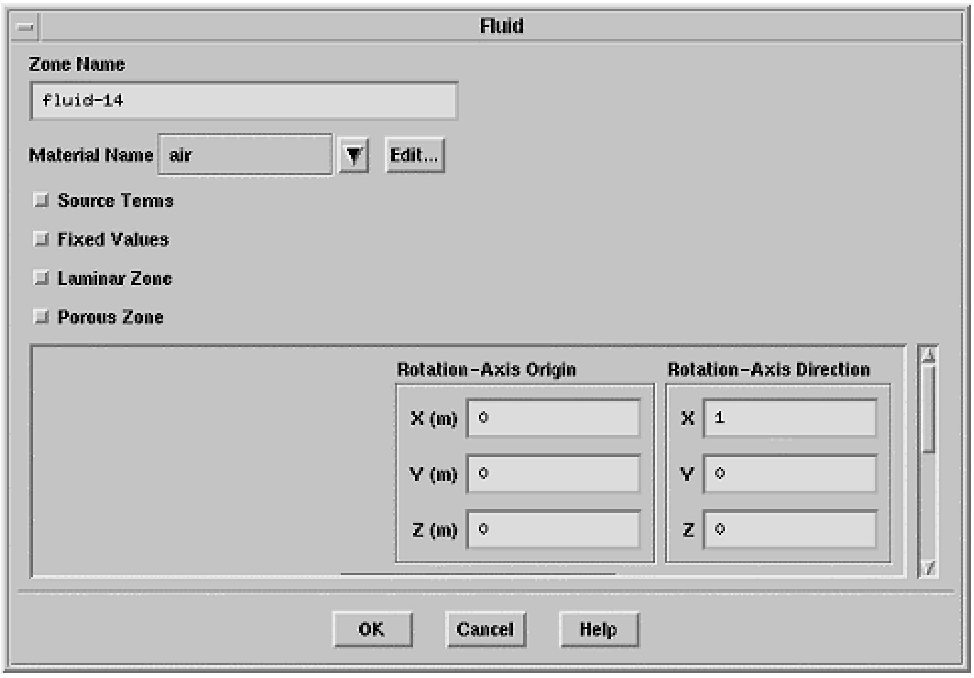


图 8-25 fluid（流体）面板

7. 定义区域的运动

在 Motion Type（运动类型）列表中选择 Moving Reference Frame（移动参考系），可以为运动的流体区域定义转动或平动的参考系。

如果想为滑移网格定义区域的运动，则可以在运动类型列表中选择 Moving Mesh（移动网格），然后完成相关参数设置。

对于平动运动则只要在 Translational Velocity（平动速度）中设定速度的 X、Y、Z 分量即可。

8. 定义辐射参数

如果计算中使用了 DO 辐射模型，可以在 Participates in Radiation（是否参与辐射）中确定流体区域是否参与辐射过程。

8.18 固体条件

固体区域是这样一类网格的集合，在这个区域上只有热传导问题被求解，与流场相关的方程则无需在此求解。被设定为“固体”的区域实际上可能是流体，只是这个流体上被假定没有对流过程发生。在固体区域上需要输入的信息只有固体的材料性质。必须指明固体的材料性质，以便计算中可以使用正确的材料信息。还可以在固体区域上设定热生成率，或固定的温度值。也可以定义固体区域的运动。如果在固体区域周围存在周期性边界，还需要指定转动轴。如果计算中使用 DO 模型计算辐射过程，还需要说明固体区域是否参与辐射过程。

8.18.1 固体区域参数的设置

所有与固体区域相关的输入在 Solid（固体）面板中完成，这个面板可以从边界条件面板中启动。

1. 定义固体材料

在材料名称（Material Name）列表中可以选择设定固体的材料，如果材料参数不符合要求，还可以通过编辑改变这些参数。

2. 定义热源

选择源项（Source Terms）就可以为固体区域设置热源。

3. 定义固定温度

在固定值（Fixed Values）选项中可以为固体区域设置一个固定的温度值。

4. 定义转动轴

转动轴仍然是通过定义轴的方向和原点位置进行定义的，在二维情况下仍然是只要确定转轴原点即可。

5. 定义区域运动

定义参考坐标系可以在 Motion Type（运动类型）列表中选择 Moving Reference Frame（运动参考系）完成定义。

定义移动网格可以在运动类型的 Moving Mesh（移动网格）中完成。

对于带有直线运动的固体区域则可以用定义 Translational Velocity（平移速度）的三个

分量来定义。

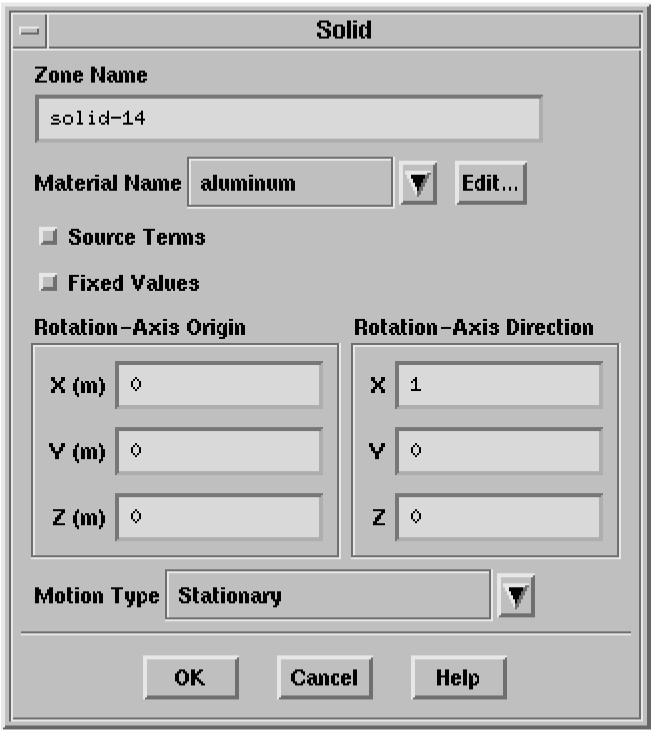


图 8-26 Solid（固体）面板

6. 定义辐射参数

如果使用 DO 模型计算辐射过程，可以在 Participates in Radiation（是否参与辐射）选项中确定固体区域是否参与辐射过程。

8.19 多孔介质条件

很多问题中包含多孔介质的计算，比如流场中包括过滤纸、分流器、多孔板和管道集阵等边界时就需要使用多孔介质条件。在计算中可以定义某个区域或边界为多孔介质，并通过参数输入定义通过多孔介质后流体的压力降。在热平衡假设下，也可以确定多孔介质的热交换过程。

在薄的多孔介质面上可以用一维假设“多孔跳跃（porous jump）”定义速度和压强的降落特征。多孔跳跃模型用于面区域，而不是单元区域，在计算中应该尽量使用这个模型，因为这个模型可以增强计算的稳定性和收敛性。

8.19.1 多孔介质模型的假设和限制条件

多孔介质模型采用经验公式定义多孔介质上的流动阻力。从本质上说，多孔介质模型就是在动量方程中增加了一个代表动量消耗的源项。因此，多孔介质模型需要满足下面的限制条件：

（1）因为多孔介质的体积在模型中没有体现，在缺省情况下，FLUENT 在多孔介质内部使用基于体积流量的名义速度来保证速度矢量在通过多孔介质时的连续性。如果希望更精确地进行计算，也可以让 FLUENT 在多孔介质内部使用真实速度，详情见 8.19.7 节。

（2）多孔介质对湍流的影响仅仅是近似。

（3）在移动坐标系中使用多孔介质模型时，应该使用相对坐标系，而不是绝对坐标系，以保证获得正确的源项解。

8.19.2 多孔介质的动量方程

在动量方程中增加一个动量源项可以模拟多孔介质的作用。源项由两部分组成：一个粘性损失项，即方程（8-45）右端第一项；和一个惯性损失项，即方程（8-45）右端第二项。

$$S_i = -\left(\sum_{j=1}^{3} D_{ij} \mu v_j + \sum_{j=1}^{3} C_{ij} \frac{1}{2} \rho v_{mag} v_j \right) \tag{8-45}$$

式中 S_i 是第 i 个（x、y 或 z 方向）动量方程中的源项，D 和 C 是给定矩阵。负的源项又被称为“汇”，动量汇对多孔介质单元动量梯度的贡献，在单元上产生一个正比于流体速度（或速度平方）的压力降。

在简单、均匀的多孔介质上，还可以使用下面的数学模型：

$$S_i = -\left(\frac{\mu}{\alpha} v_i + C_2 \frac{1}{2} \rho v_{mag} v_i \right) \tag{8-46}$$

式中 α 代表多孔介质的渗透性，C_2 是惯性阻力因子，将 D 和 C 分别定义为由 $1/\alpha$ 和 C_2 为对角单元的对角矩阵。

FLUENT 中还可以用速度的指数律作为源项的模型，即：

$$S_i = -C_0 |v|^{C_1} = -C_0 |v|^{(C_1 - 1)} v_i \tag{8-47}$$

式中 C_0 和 C_1 为用户自定义的经验常数。其中压力降是各向同性的，C_0 的单位为国际

单位制。

1. 多孔介质的 Darcy 定律

在流过多孔介质的层流中，压力降正比于速度，常数C_2可以设为零。忽略对流加速和扩散项，多孔介质就简化为 Darcy 定律：

$$\nabla p = -\frac{\mu}{\alpha}\vec{v} \tag{8-48}$$

FLUENT 在 x、y、z 三个坐标方向计算出的压力降为：

$$\Delta p_x = \sum_{j=1}^{3}\frac{\mu}{\alpha_{xj}}v_j\Delta n_x$$

$$\Delta p_y = \sum_{j=1}^{3}\frac{\mu}{\alpha_{yj}}v_j\Delta n_y \tag{8-49}$$

$$\Delta p_z = \sum_{j=1}^{3}\frac{\mu}{\alpha_{zj}}v_j\Delta n_z$$

式中的$1/\alpha_{ij}$就是方程（8-45）中的D，v_j是 x、y、z 三个坐标方向的速度分量，Δn_x、Δn_y、Δn_z是多孔介质在 x、y、z 三个坐标方向的真实厚度。如果计算中使用的厚度值不等于真实厚度值，则需要对$1/\alpha_{ij}$做出调整。

2. 多孔中的惯性损失

在流速很高时，方程（8-45）中的常数C_2可以对惯性损失做出修正。C_2可以被看作流动方向上单位长度的损失系数，这样一来就可以将压力降定义为动压头的函数。

如果计算的是多孔板或管道阵列，在一些情况下可以略去渗透项，而只保留惯性损失项，并产生下面简化形式的多孔介质方程：

$$\nabla p = -\sum_{j=1}^{3}C_{2ij}\left(\frac{1}{2}\rho v_j v_{mag}\right) \tag{8-50}$$

或者写成分量形式：

$$\Delta p_x \approx \sum_{j=1}^{3} C_{2xj}\Delta n_x \frac{1}{2}\rho v_j v_{mag}$$

$$\Delta p_y \approx \sum_{j=1}^{3} C_{2yj}\Delta n_y \frac{1}{2}\rho v_j v_{mag} \qquad (8\text{-}51)$$

$$\Delta p_z \approx \sum_{j=1}^{3} C_{2zj}\Delta n_z \frac{1}{2}\rho v_j v_{mag}$$

再次说明，上式中的厚度即模型中定义的厚度。

8.19.3 多孔介质能量方程的处理

多孔介质对能量方程的影响体现在对对流项和时间导数项的修正上。在多孔介质对对流项的计算中采用了有效对流函数，在时间导数项中则计入了固体区域对多孔介质的热惯性效应：

$$\frac{\partial}{\partial t}\left(\gamma\rho_f E_f + (1-\gamma)\rho_s E_s\right) + \nabla\cdot\left(\vec{v}\left(\rho_f E_f + p\right)\right) = \nabla\cdot\left[k_{eff}\nabla T - \left(\sum_i h_i J_i\right) + \left(\bar{\tau}\cdot\vec{v}\right)\right] + S_f^h$$

（8-52）

式中 E_f 为流体总能，E_s 为固体介质总能，γ 为介质的多孔性，k_{eff} 介质的有效热导率，S_f^h 流体焓的源项。

FLUENT 中使用的有效导热率 k_{eff} 是流体导热率和固体导热率的体积平均值：

$$k_{eff} = \gamma k_f + (1-\gamma)k_s \qquad (8\text{-}53)$$

式中 γ 为介质的多孔率，k_f 为流体的热导率，k_s 为固体的热导率。k_f 和 k_s 可以用自定义的函数计算。各向异性的热导率也可以用自定义函数定义。

8.19.4 多孔介质模型对湍流的处理

在缺省情况下，FLUENT 在多孔介质计算中通过求解标准守恒型方程计算湍流变量。在计算过程中，通常假设固体介质对湍流的生成和耗散没有影响。在多孔介质的渗透率很大，因而介质的几何尺度对湍流涡结构没有影响时，这个假设是合理的。在另外一些情况

中，可能不需要考虑流动中的湍流，即假定流动为层流。

如果计算中使用的湍流模型是 $k-\varepsilon$ 、$k-\omega$ 或 Spalart-Allmaras 模型中的一种，则可以通过将湍流粘度 μ_t 设为零的方式忽略湍流的影响。如果湍流粘度 μ_t 被设置为零，则在计算过程中仍然会将湍流变量输运到介质的另一面，但是湍流对动量输运过程的影响则完全被消除。在 Fluid（流体）面板中将多孔介质区设为 Laminar Zone（层流区）选项就可以将湍流粘度 μ_t 设为零。

8.19.5 多孔介质对瞬态标量方程的影响

对于多孔介质的瞬态计算，多孔介质对时间导数项的影响体现在对所有变量的输运方程和连续性方程求解中。在考虑多孔介质的影响时，时间导数项变为 $\frac{\partial}{\partial t}(\gamma\rho\phi)$，式中 ϕ 为任意流动变量，γ 为多孔率。在瞬态流动计算中，多孔介质的影响是被自动加入计算过程的，其多孔率在缺省设置中等于 1。

8.19.6 多孔介质计算中用户的输入参数

多孔介质计算中需要输入的项目如下：

（1）定义多孔介质区域。
（2）定义多孔介质速度函数形式。
（3）定义流过多孔介质区的流体属性。
（4）设定多孔区的化学反应。
（5）设定粘性阻力系数。
（6）设定多孔介质的多孔率。
（7）在计算热交换的过程中选择多孔介质的材料。
（8）设定多孔介质固体部分的体热生成率。
（9）设定流动区域上的任意固定值的流动参数。
（10）需要的话，将多孔区流动设为层流，或取消湍流计算。
（11）定义旋转轴或区域的运动。

所有参数设置均在流体（Fluid）面板中完成，见图 8-27。

1. 定义多孔区

选中 Porous Zone（多孔介质区）选项就可以将流体区域设为多孔介质。



图 8-27 多孔计算选项

2. 定义多孔介质速度公式

在 Solver（求解器）面板中有一个 Porous Formulation（多孔公式）区可以确定在多孔介质区域上使用名义速度或物理速度。缺省设置为名义速度。

3. 定义流过多孔介质的流体

在 Material Name（材料名称）中选择所需的流体名称即可。可以用编辑功能改变流体的参数设置。组元计算或多相流计算中的流体不在这里定义，而是在 Species Model（组元模型）面板中定义。

4. 在多孔区域上设置化学反应

在 Fluid（流体）面板上选中 Reaction（反应）选项，再从 Reaction Mechanism（反应

机制）中选择合适的反应就可以在多孔介质区域的计算中加入化学反应。

如果化学反应中包括表面反应，则需要设定 Surface to Volume Ratio（面体比）。面体比是多孔介质单位体积上拥有的表面积，因此可以作为催化反应强度的度量。根据这个参数，FLUENT 可以计算出体积单元上总的表面积。

5. 定义粘性和惯性阻力系数

粘性和惯性阻力系数的定义方式是相同的。在直角坐标系中定义阻力系数的办法是：在二维问题中定义一个方向矢量，或在三维问题中定义两个方向矢量，然后再在每个方向上定义粘性和惯性阻力系数。在二维计算中的第二个方向，即没有被显式定义的那个方向，是与被定义的方向矢量相垂直的方向。与此类似，在三维问题中的第三个方向为垂直于前两个方向矢量构成平面的方向。在三维问题中，被定义的两个方向矢量应该是相互垂直的，如果不垂直的话，FLUENT 会将第二个方向矢量中与第一个方向矢量平行的分量删除，强制令二者保持垂直。因此第一个方向矢量必须准确定义。

用 UDF 也可以定义粘性和惯性阻力系数。在 UDF 被创建并调入 FLUENT 后，相关的用户定义选项就会出现在下拉列表中。需要注意的是，用 UDF 定义的系数必须使用 DEFINE_PROPERTY 宏。

如果计算的问题是轴对称旋转流，可以为粘性和惯性阻力系数定义一个附加的方向分量。这个方向应该与其他两个方向矢量相垂直。

在三维问题中，还允许使用圆锥（或圆柱）坐标系。需要提醒的是，多孔介质流中计算粘性或惯性阻力系数时采用的是名义速度。

6. 定义阻力的步骤

定义阻力系数的步骤如下：

（1）定义方向矢量。

1）如果采用直角坐标系，在二维计算中只要定义第一方向矢量（Direction-1 Vector）即可，在三维计算中则还要定义第二方向矢量（Direction-2 Vector）。矢量的起点都是（0,0）或（0,0,0），终点则用坐标（x,y）或（x,y,z）定义。

在无法确定多孔介质的方向矢量时，可以使用三维的平面工具或二维的线工具定义方向矢量：

- 在多孔介质边界上启动 plane tool（面工具）或 line tool（线工具）。
- 旋转面工具或线工具的轴，使之与多孔介质相吻合。
- 从 Fluid（流体）面板中点击 Update From Plane Tool（从面工具更新）或 Update From Line Tool（从线工具更新），FLUENT 就会将工具的红箭头方向设为第一方向矢量。在三维问题中，同时将绿色箭头方向设为第二方向矢量。

2）在三维问题中还可以使用圆锥坐标系完成设置，具体步骤如下：

- 选择 Conical（圆锥）选项。
- 定义 Cone Axis Vector（圆锥轴矢量）和 Point on Cone Axis（圆锥轴上的点）。

圆锥轴矢量的方向由起点为（0,0,0）终点为（x,y,z）的矢量确定。而圆锥轴上的点则用于将阻力转换为直角坐标系中的值。

- 设置 Cone Half Angle（半锥角）。如果使用的是圆柱坐标系，则将半锥角设为零。在事先不知道圆锥轴的方向和锥面上一点坐标的时候，可以用面工具设置圆锥轴矢量和锥面上一点的坐标。一种方法是：

  （a）在 Snap to Zone（转换到区域）按钮旁边选择垂直于圆锥轴的边界区域。

  （b）点击转换到区域按钮，FLUENT 会自动将面工具设在边界上。同时设定的有 Cone Axis Vector（圆锥轴矢量）、Point on Cone Axis（圆锥轴上一点）和半锥角。

  另一种方法是：

  （a）将面工具设在多孔区域上。

  （b）转动、平移面工具的轴，使得红色箭头指向圆锥轴的方向，同时使面工具的原点与圆锥轴的原点重合。

  （c）在二者的轴和原点全部重合后，点击 Update From Plane Tool（从面工具更新）按钮，FLUENT 就会自动设定圆锥轴矢量和圆锥上一点的坐标。

（2）在 Viscous Resistance（粘性阻力）下，在每个方向上定义粘性阻力系数$1/\alpha$，在 Inertial Resistance（惯性阻力）下，在每个方向上定义惯性阻力系数C_2。如果使用圆锥定义方法，则方向一（Direction-1）为圆锥轴方向，方向二（Direction-2）为垂直于锥面的方向，对应于圆柱坐标系中的径向，方向三（Direction-3）为周向。

在三维问题中有三类系数，二维问题中则有两类：

1）在各向同性问题中，所有方向的阻力系数是相同的，比如海绵。在这类问题中，必须明确地将各方向的阻力系数设为同一个值。

2）在某个方向的阻力系数与其他方向有所不同时，应该谨慎设置阻力参数。比如，如果一个多孔介质区由圆柱形吸管组成，并且吸管的开口方向与流动方向平行，则流体会比较容易地流过吸管，但是在其他两个方向则不易流过。如果有一个与流动方向垂直的平板，则流体完全不能流过平板，而是沿平板向其他两个方向流动。

3）三维问题中另一种可能情况是所有三个方向都是不同的。比如，如果多孔介质区由不规则排列的物体组成，则流体在其中流过时在各个方向上的过程都是不同的。因此需要在各个方向上设置不同的系数。

7. 粘性和惯性阻力系数的推导

推导粘性和惯性阻力系数的方法如下：

（1）采用已知的压强损失，基于名义速度推导多孔介质参数。

使用多孔介质模型必须注意的是，FLUENT 中假设多孔介质单元是完全浸没在流体中的，而用户定义的$1/\alpha_{ij}$和C_{2ij}都是基于这个假设得出的。假定真实容器中的压强降与速度

的关系是已知的，而其中的多孔介质却是部分浸没在流体中的。通过下面的练习可以掌握一种计算C_2的方法。

例题：一块多孔板有 25％浸没在流体中，压强降为动压头的 0.5 倍，损失因子K_L的定义为：

$$\Delta p = K_L \left(\frac{1}{2} \rho v_{25\%}^2 \right) \tag{8-54}$$

因此K_L应该等于 0.5，计算动压使用的速度是经过 25％浸没在流体中的多孔介质上的速度。计算C_2时应该注意：

1）在假定多孔板 100％浸没在流体中的前提下计算其速度。
2）损失系数必须被转化为单位长度上动压损失。

因此损失因子K'_L适用于下面的公式：

$$\Delta p = K'_L \left(\frac{1}{2} \rho v_{100\%}^2 \right) \tag{8-55}$$

式中$v_{25\%} = 4 \times v_{100\%}$，

$$K'_L = K_L \times \frac{v_{25\%}^2}{v_{100\%}^2} = 0.5 \times \left(\frac{4}{1} \right)^2 = 8 \tag{8-56}$$

根据第 2）条要求，还需要将K'_L转换为单位厚度多孔板上的损失系数。假定板的厚度为 1mm，则惯性损失因子为：

$$C_2 = \frac{K'_L}{厚度} = \frac{8}{10^{-3}} = 8000 m^{-1} \tag{8-57}$$

对各向异性介质，上述过程需要在各坐标方向上重复进行。

（2）用 Ergun 方程计算充填床的多孔介质参数。

作为第二个例子，可以研究一下充填床问题。在湍流中，充填床的数学模型是用穿透率和惯性损失系数来定义的。计算相关常数的一种办法是使用半经验公式 Ergun 方程，这个方程适用的雷诺数范围很广，同时也使用于多种填充物：

$$\frac{|\Delta p|}{L}=\frac{150\mu}{D_p^2}\frac{(1-\varepsilon)^2}{\varepsilon^3}v_\infty+\frac{1.75\rho}{D_p}\frac{(1-\varepsilon)}{\varepsilon^3}v_\infty^2 \qquad (8\text{-}58)$$

在计算层流时，（8-58）式中右端第二项可以被去掉，Ergun 方程随之简化为 Blake-Kozeny 方程：

$$\frac{|\Delta p|}{L}=\frac{150\mu}{D_p^2}\frac{(1-\varepsilon)^2}{\varepsilon^3}v_\infty \qquad (8\text{-}59)$$

在上述方程中 μ 为粘度，D_p 为粒子的平均直径，L 为床的深度，ε 为空腔比率，定义为空腔与填充床的体积比。对比方程（8-48）、（8-50）和（8-58）可知，各方向上的渗透率和惯性损失系数为：

$$\alpha=\frac{D_p^2}{150}\frac{\varepsilon^3}{(1-\varepsilon)^2} \qquad (8\text{-}60)$$

$$C_2=\frac{3.5}{D_p}\frac{(1-\varepsilon)}{\varepsilon^3} \qquad (8\text{-}61)$$

（3）用经验公式推导穿过多孔板的流动参数

作为第三个例子，下面用 Van Winkle 方程计算带方孔的多孔板上压强的损失。方程的提出者认为该方程适用于呈三角形分布的等距方孔板的湍流计算，具体形式如下：

$$\dot{m}=CA_f\sqrt{(2\rho\Delta p)/\left(1-(A_f/A_p)^2\right)} \qquad (8\text{-}62)$$

式中 $\dot{m}$ 为通过板的质量流量，A_f 为孔的总面积，A_p 板的总面积（固体与孔的和），D/t 孔直径与板厚之比，C 是随雷诺数和 D/t 变化的系数，其值可以通过查表获得。在 $t/D>1.6$，且 $\mathrm{Re}>4000$ 时，C 近似等于 0.98，其中雷诺数是用孔的直径做特征长，孔中流体的速度做特征速度求出的。

将方程 $\dot{m}=\rho v A_p$ 代入（8-62），并除以板厚 $\Delta x=t$，可得：

$$\frac{\Delta p}{\Delta x}=\left(\frac{1}{2}\rho v^2\right)\frac{1}{C^2}\frac{(A_p/A_f)^2-1}{t} \qquad (8\text{-}63)$$

其中 v 是名义速度。与方程（8-50）对比后可以发现：

$$C_2=\frac{1}{C^2}\frac{(A_p/A_f)^2-1}{t} \qquad (8\text{-}64)$$

（4）用表格数据计算流过纤维垫的多孔介质参数

在计算流体流过纤维垫或过滤器这类问题时，除了 Blake-Kozeny 方程外，还可以用试验数据进行计算，见下表：

表 8-6 纤维计算试验数据表格

固体材料的体积比	纤维的无量纲渗透率 B
0.262	0.25
0.258	0.26
0.221	0.40
0.218	0.41
0.172	0.80

表中 $B = \alpha / a^2$， a 为纤维直径。α 的含义与（8-48）式中相同，对于给定纤维直径和体积比的多孔介质是容易求出的。

（5）用压强和速度的试验数据计算多孔介质系数

在已知多孔介质上的速度与压强降的试验数据时，可以通过插值求出多孔介质上的系数。下面通过一个例子说明这一点。假设试验数据如表 8-7 所示。

采用表 8-7 的数据可以拟合出一条“速度－压强降”曲线，其方程为：

$$\Delta p = 0.28296 v^2 - 4.33539 v \qquad (8\text{-}65)$$

其中 Δp 为压强降， v 为速度。这个方程等价于方程（8-46），形式上是速度 v 的多项式，对比两个方程可知：

$$0.28296 = C_2 \frac{1}{2} \rho \qquad (8\text{-}66)$$

表 8-7 速度、压强降的试验数据

速度（m/s）	压强降（Pa）
20.0	78.0
50.0	487.0
80.0	1432.0
110.0	2964.0

在 $\rho = 1.225 kg / m^3$ 时，惯性阻力因子 $C_2 = 0.462$ 。同样地，

$$-4.33539 = \frac{\mu}{\alpha} \qquad (8\text{-}67)$$

在 $\mu = 1.7894 \times 10^{-5}$ 时，可知 $1/\alpha = -242282$ 。

8. 采用指数律模型

如果采用指数律模型计算多孔介质的动量源项，则需要输入的系数只有C_0和C_1。在Fluid（流体）面板的 Power Law Modal（指数律模型）下，输入C_0和C_1。

指数律可以与 Darcy 和惯性模型同时使用，C_0必须使用国际单位制。

9. 定义多孔性

在 Fluid（流体）面板中的 Fluid Porosity（流体多孔率）下设置多孔率，即可定义计算中的多孔率参数。

定义多孔率的另一个方法是使用 UDF 函数。在创建了相关函数并将其载入 FLUENT 后就可以在计算中使用了。

10. 定义多孔介质材料

在 Fluid（流体）面板中的 Fluid Porosity（流体多孔率）下，选择 Solid Material Name（固体材料名称），然后直接进行选择即可。如果固体材料的属性参数不符合计算要求，可以对其进行编辑，比如可以用 UDF 函数编辑材料的各向异性导热率。

11. 定义源项

如果在计算中需要考虑多孔介质上的热量生成，可以开启 Source Terms（源项）选项，并设置一个非零的 Energy（能量）源项。求解器将把用户输入的源项值与多孔介质的体积相乘获得总的热量生成量。

12. 定义固定值

如果有些变量的值不需要由计算得出，就可以选择 Fixed Values（固定值）选项，并认为设定这些参数。

13. 在多孔区域中取消湍流计算

在 Fluid（流体）面板中，开启 Laminar Zone（层流区）选项，就可以将湍流粘度设为零，从而使相关区域中的流动保持层流状态。

14. 定义旋转轴和区域运动

其方法与标准流体区域上的设置相同，这里不再重复。

8.19.7 用物理速度定义多孔介质

在多孔介质的计算中，FLUENT 在缺省情况下用名义速度计算多孔介质区域的流动。名义速度可以在总体上保证流动的连续性，但是对于流动细节的考虑则不如物理速度，因此如果要更准确地计算多孔介质流动应该使用物理速度。名义速度与物理速度的关系如下：

$$\vec{v}_{名义} = \gamma \vec{v}_{物理} \qquad (8\text{-}68)$$

在 Solver（求解器）面板中的 Porous Formulation（多孔介质公式）中可以定义物理速度。

采用物理速度公式，假定 ϕ 为通用的流体变量，则各向同性多孔介质的控制方程为：

$$\frac{\partial(\gamma\rho\phi)}{\partial t} + \nabla\cdot(\gamma\rho\vec{v}\phi) = \nabla\cdot(\gamma\Gamma\nabla\phi) + \gamma S_\phi \qquad (8\text{-}69)$$

假定流动为单相流，则体积平均的质量和动量应满足的守恒型方程如下：

$$\frac{\partial(\gamma\rho)}{\partial t} + \nabla\cdot(\gamma\rho\vec{v}) = 0 \qquad (8\text{-}70)$$

$$\frac{\partial(\gamma\rho\vec{v})}{\partial t} + \nabla\cdot(\gamma\rho\vec{v}\vec{v}) = -\gamma\nabla p + \nabla\cdot(\gamma\bar{\tau}) + \gamma\vec{B}_f - \left(\frac{\mu}{\alpha} + \frac{C_2\rho}{2}|\vec{v}|\right)\vec{v} \qquad (8\text{-}71)$$

（8-71）式的最后一项是由多孔介质产生的粘性和惯性阻力项。

8.19.8 多孔介质流动的求解策略

在多数情况下，可以用标准的求解次序计算多孔介质流动，但是在计算过程中可以发现计算的收敛速度大大放慢，特别是通过多孔介质区的压强降很大时更是如此。收敛速度放慢的主要原因是多孔介质引起的压强降在动量方程中是作为一个源项出现的，而源项则可以看作对计算增加了一个扰动，使原有格式的收敛性有所降低。

解决收敛性问题的方法是提高初始流场的准确性，初始流场的准确性可以加速收敛过程。比如在多孔介质前后设置更接近真实压强值的压强条件，或者在计算中先用不带多孔介质模型的模型计算出初始流场，再把多孔介质模型设置到计算中，这样在一个更接近于最终结果的初始流场中开始的计算能够更快地收敛。

在各向异性多孔介质的计算中，如果某个方向上的阻力系数远远大于其他两个方向，也无需将阻力因子设置成很大的值，而最好是将它限制在其他两个方向的阻力系数的 2、3 个量级以内。比如多孔介质的一个方向上的阻力系数可能是无穷大，但是在计算中不要将阻力系数设置成无穷大，而是需要将它限制在主流方向阻力系数的 1000 倍以内。

8.19.9 多孔介质的后处理

能够反映多孔介质流动特点的参数是速度和压强。这两个参数在 FLUENT 内建的后处理模块中都可以找到，并很容易进行图形显示等操作。

需要注意的是多孔介质的热力学变量的计算结果不能反映固体介质的性质。报告中显示的密度、热容、热导率和焓，都是多孔介质区域中流体的属性，与固体介质无关。

如果多孔介质流动计算中存在表面化学反应，可以用 Reactions...（反应）项中的 Arrhenius Rate of Reaction-n（第 n 个反应的 Arrhenius 反应速率）来表示。

8.20 风扇边界条件

在已知风扇几何特征和流动特征的条件下，风扇的这些特征可以被参数化用于计算风扇风扇对流场的影响。在风扇边界条件中可以输入一条确定风扇前后压力头与速度关系的经验曲线。同时可以确定的量还包括风扇旋转速度的径向与周向速度分量。风扇模型不是绕风扇流动的精确模型，但是它可以计算流过风扇的流量。风扇可以与其他类型的源项共同使用，也可以作为唯一的源项使用。在作为唯一源项时，系统流量的计算是在考虑系统损失和前面提到的风扇曲线的前提下完成的。

8.20.1 风扇方程

1. 经过风扇的压强跃升模型

风扇的物理模型是一个无限薄的面，压强在经过这个面时出现跃升，而跃升的值是速度的函数。

在函数关系取为多项式的情况下，压强跃升函数的形式为：

$$\Delta p = \sum_{n=1}^{N} f_n v^{n-1} \quad （8\text{-}72）$$

式中 Δp 为压强跃升量，f_n 为多项式系数，v 为速度。需要注意的是在这个关系式中，速度 v 的值应该保持为正值，因为这代表流体是向前流过风扇，所以才会产生压强跃升。另一个确定压强跃升的方法是用垂直于风扇的质量平均速度作为计算压强跃升的速度，算出风扇区域的压强跃升。

2. 风扇旋转速度的计算

三维问题中，可以在风扇表面定义径向和周向对流速度以模拟旋转流动。速度分量可以作为径向距离的函数进行定义。在输入旋转速度时必须使用国际单位制。

在用多项式定义径向和周向速度时，其函数形式如下：

$$U_\theta = \sum_{n=-1}^{N} f_n r^n; \qquad -1 \le N \le 6 \qquad (8\text{-}73)$$

$$U_r = \sum_{n=-1}^{N} g_n r^n; \qquad -1 \le N \le 6 \qquad (8\text{-}74)$$

式中U_θ和U_r为风扇表面的周向和径向速度，f_n和g_n为多项式系数，r为径向距离。

8.20.2 风扇计算的输入项

风扇边界条件在 Fan（风扇）面板中输入，这个面板可以从 Boundary Conditions（边界条件）面板中进入。

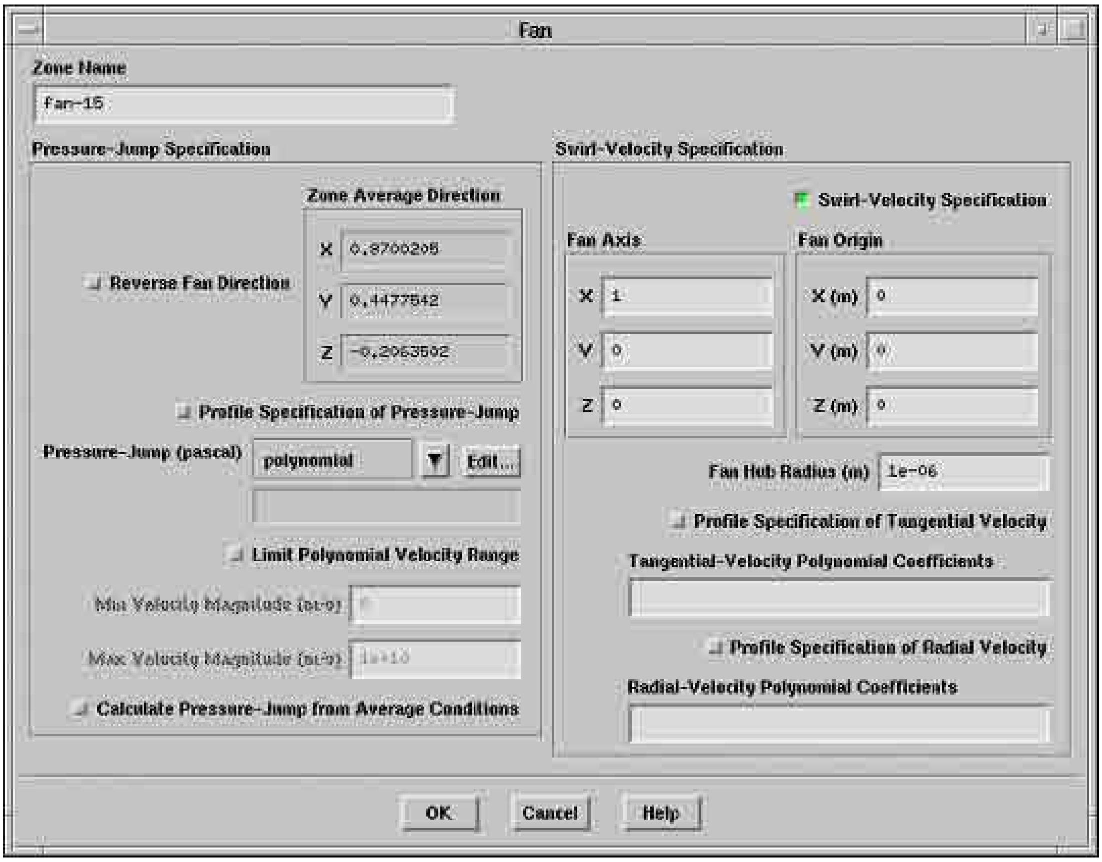


图 8-28 风扇边界条件

风扇边界的输入项包括：

（1）定义风扇区域。

（2）定义压强跃升。

（3）定义风扇的离散项边界条件。

（4）定义旋转速度。

1. 定义风扇区域

因为在计算中假定风扇为无限薄的一个面，所以必须被定义为单元之间的一个面，而不是一个单元。在二维问题中，风扇边界应该是一条线；在三维问题中，风扇边界应该是一个三角或四边形单元。如果在读入网格的时候，风扇区域被定义为内部区域，可以用 Boundary Conditions（边界条件）面板将其类型改变为 fan（风扇）区域类型。

在将内部区域类型改变为风扇类型后，就可以打开风扇面板定义压强跃升和旋转速度了。

2. 定义压强跃升

与其他变量的定义一样，压强跃升的定义也可以采用常数、多项式、阶跃函数、分段多项式和用户自定义函数（UDF）来定义，其操作过程也是类似的。

在假设压强跃升为常数时，可以在 Fan（风扇）面板中去掉 Profile Specification of Pressure-Jump（压强跃升的型函数定义）选项，在下面的栏目中填入压强跃升的值即可。

在使用函数定义时，则选中压强跃升的型函数选项，然后在 Pressure-Jump（压强跃升）右端选择函数类型，点击 Edit（编辑）按钮编辑函数形式就可以了。

在使用上述函数形式定义压强跃升时，可以定义速度的最大值和最小值。打开 Limit Polynomial Velocity Range（多项式速度范围极限）选项，可以在 Max Velocity Magnitude（最大速度值）和 Min Velocity Magnitude（最小速度值）中填入速度的上下两个极限值。在计算过程中，如果速度超过了最大速度值，则速度被自动设定为最大速度值。同样，如果速度值小于最小速度值，则速度被自动设定为最小速度值。

3. 定义风扇边界上的弥散相边界条件

弥散相边界条件在弥散相面板中输入，详情可以参考后面相关章节。

4. 定义风扇旋转速度

应该按下列顺序为风扇设定径向和周向速度：

（1）打开 Swirl-Velocity Specification（旋转速度说明）。

（2）定义风扇的转轴和原点。

（3）设定 Fan Hub Radius（桨毂半径）。

（4）用常数、多项式函数或 UDF 函数设定径向与周向速度函数。

注意必须使用国际单位制定义所有旋转速度分量。函数设定具体方法如下：

- 多项式形式速度函数设定

  1）打开 Profile Specification of Tangential Velocity（周向速度分布函数定义）或 Profile Specification of Radial Velocity（径向速度型函数定义）选项。

  2）在下面的多项式系数输入栏中，按由小到大的次序输入多项式系数 f_n 或 g_n。

  每个系数之间用空格分隔开来。

- 速度值为常数的输入

  1）打开 Profile Specification of Tangential Velocity（周向速度分布函数定义）或 Profile Specification of Radial Velocity（径向速度分布函数定义）选项。

  2）在下拉列表中选择 constant（常数）。

  3）输入周向或径向的速度值。

  更为简便的方法是：

  1）打开 Profile Specification of Tangential Velocity（周向速度分布函数定义）或

Profile Specification of Radial Velocity（径向速度分布函数定义）选项。

2）在下面的多项式系数输入栏中，直接输入相应的速度值。

8.20.3 风扇计算的后处理

1. 压强跃升报告

可以用 Surface Integrals（表面积分）面板报告压强跃升，步骤如下：

（1）在风扇区的两面创建表面。用 Transform Surface（转换面）面板将风扇区域略向上游和下游移动形成两个新的表面。

（2）在表面积分面板中，报告风扇上下游的平均静压。然后就可以计算出风扇前后的压强跃升。

2. 图形显示

与风扇计算相关的图形显示有：

（1）静压和静温的等值线或型线图。

（2）静压或静温沿空间位置的分布曲线。

具体的操作方法参见与后处理技术相关的章节。

8.21 散热器边界条件

FLUENT 可以计算换热器件（如散热器或冷凝器）的流场。散热器边界条件是将压强降和热交换系数作为散热器法向速度的函数定义其数学模型的。

8.21.1 散热器方程

1. 散热器的压强损失

散热器被假定为一个无限薄的面，经过这个面的压强降与流体动压成正比，其比例系数为一个经验系数：

$$\Delta p = k_L \frac{1}{2} \rho v^2 \qquad (8\text{-}75)$$

式中 ρ 为流体密度，k_L 是无量纲损失系数。

2. 通过散热器的热交换

从散热器向周围流体的热流通量为：

$$q = h\left(T_{air,d} - T_{ext}\right) \qquad (8\text{-}76)$$

式中 q 为热流通量，$T_{air,d}$ 为散热器下游温度，T_{ext} 是流体的参考温度。对流热交换系数 h 可以用常数、多项式、阶跃函数和分段多项式来表示。对流热交换系数 h 可以用上下游温度和散热器面积计算得出。

8.21.2 散热器的参数定义

散热器参数定义是在 Radiator（散热器）面板中完成的。

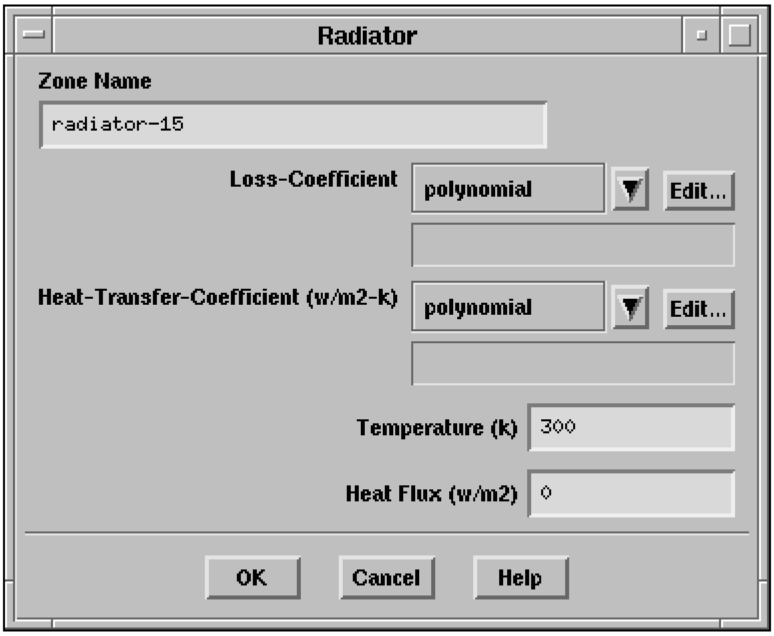


图 8-29 散热器面板

在散热器计算中需要输入的参数包括：

（1）定义散热器区域。

（2）定义压强损失系数。

（3）定义热流通量、热交换系数和散热器温度。

（4）如果计算中包含弥散相问题，还需定义弥散相边界。

1. 定义散热器区域

散热器区域为内部边界面。在 FLUENT 读入网格文件时，散热器区域被定义为 interior（内部）区域。在读入网格文件后，需要在 Boundary Conditions（边界条件）面板中将它重新定义为 radiator（散热器）区域，然后再在 Radiator（散热器）面板中定义损失系数和热流通量信息。

2. 定义压强损失系数函数

压强损失系数 k_L 也可以用常数、多项式、阶跃函数、分段多项式等四种形式进行定义。其定义方式与其他类似参数定义过程相类似，这里不再重复。

3. 定义热流通量参数

定义热流通量参数既可以定义真实的热流通量 q，也可以设置热交换系数或散热器温度。所有设置过程均在散热器面板中完成。

如果采用定义真实的热流通量，先将 Temperature（温度）设为零，然后将 Heat Flux（热流通量）设置为一个常数。

如果选择定义散热器温度，可以在 Temperature（温度）栏中输入外界温度的值。如果要定义热交换系数，可以在常数、多项式、阶跃函数、分段多项式等方式中选择一种进行设置。

4. 定义弥散相边界条件

具体方法请参见弥散相计算的相关章节。

8.21.3 散热器计算的后处理

1. 散热器压强降的报告

可以在 Surface Integrals（表面积分）面板中报告经过散热器的压强降，具体步骤如下：

（1）在散热器前后创建表面，然后用 Transform Surface（表面转换）面板将散热器区域向上下游稍微移动一段距离，从而创建两个新的表面。

（2）在表面积分面板中，报告上下游的 Static Pressure（静压）分布，然后就可以计算出散热器前后的压强降。

如果想检验压强降的值是否正确，可以让系统先报告法向速度的值，然后再用公式（8-75）进行验算。

2. 散热器热交换参数报告

如果要计算流体经过散热器以后温度的增量，可以采用报告压强降的办法进行计算，即首先显示报告散热器上下游静温的值，然后再计算得出静温的增量。

3. 图形输出

在散热器计算中，可以绘制静压和静温的等值线图，也可以绘制静压和静温沿一个坐标方向的变化曲线。

8.22 多孔跃升边界条件

在已知一个肋板前后的速度或压强的增量时，可以用多孔跃升边界对这个肋板进行定义。多孔跃升模型比多孔介质模型简单，采用这种模型计算过程将更强健，收敛性更好，更不容易在扰动下发散，因此在计算过滤器、薄肋板等内部边界时应该尽量采用这种边界条件。

有限厚度的多孔介质的压强变化是用 Darcy 定律和一个附加的惯性损失项来定义的：

$$\Delta p = -\left(\frac{\mu}{\alpha}v + C_2\frac{1}{2}\rho v^2\right)\Delta m \qquad (8\text{-}77)$$

式中 μ 是层流粘度，α 是渗透率，C_2 是压强跃升系数，v 是法向速度，Δm 是介质厚度。α 和 C_2 可以用 8.19.6 小节中的方法计算得出。

1. 多孔跃升模型的参数输入

多孔跃升模型的相关参数在 Porous Jump（多孔跃升）面板中输入，见图 8-30。需要输入的参数如下：

（1）定义多孔跃升区域。

（2）设定面的渗透率（Face Permeability），即设定 α 的值。

（3）设定多孔介质的厚度 Δm 。

（4）设定压强跃升系数 C_2 。

（5）如果需要考虑弥散相在多孔跃升区域定义弥散相边界条件。

多孔跃升模型是对多孔介质模型的一维简化，因此就象多孔介质模型一样，是应用在无厚度的内部面上的。在使用多孔跃升模型时定义弥散相边界条件的方法请参见与弥散相计算相关的章节。

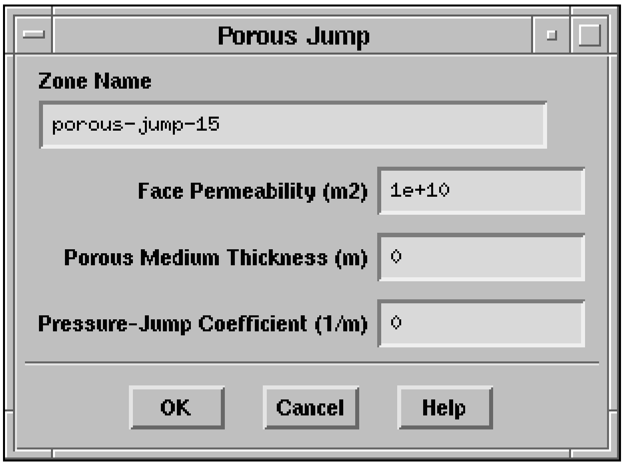


图 8-30 多孔跃升面板

2. 多孔跃升模型计算的后处理

多孔跃升模型计算的后处理内容和方法与多孔介质模型相同，具体内容请参见 8.19.9 小节。

8.23 无反射边界条件

8.23.1 概述与限制条件

在可压流计算中，特别是在内流计算中，如果将边界全部设置为无滑移条件的话，有可能使得计算中产生的一些虚假波动在壁面上被重新反射回计算域，从而导致计算的振荡，无法获得收敛解。这种情况下一般采用无反射边界条件，让波动透射过边界，以便更快速地获得收敛解。

采用无反射边界条件有下列限制条件：

（1）只能与耦合求解器配合使用。

（2）FLUENT6.1 中仅用于定常可压流计算，密度满足理想气体假设。

（3）入口和出口边界必须采用压强入口和出口条件，同时入口和出口必须使用圆柱坐标系。

（4）三维问题中，在入口和出口边界上必须使用结构化的表面网格。

（5）FLUENT 中的无反射边界条件是从二维计算中发展过来的，因此在三维计算中实际上使用的是一种伪三维形式，即用二维方法计算三维问题，其实现途径是将周向速度从三维计算中解耦出来单独进行计算。

8.23.2 使用无反射边界条件

如果要采用无反射边界条件，最好先在不使用这类边界条件时将问题计算一遍，在获得收敛解之后，再加入无反射边界条件，继续进行计算并再度获得收敛解。使用无反射边界条件的步骤如下：

（1）加入无反射边界条件的文本命令如下：

define->boundary-conditions->non-reflecting->enable?

如果不知道无反射边界条件是否已经加入计算可以用文本命令 show-status 进行查看。

（2）无反射边界条件初始化的文本命令如下：

define->boundary-conditions->non-reflecting->initialize

初始化成功后，系统会显示相应的系统信息。

（3）如果有必要，可以修改相关参数，相关的文本命令如下：

define->boundary-conditions->non-reflecting->set

相关参数的含义为：

- under-relaxation：设定亚松弛因子，缺省值为 0.75。
- discretization：设定离散格式，缺省为高阶格式。
- verbosity：设定信息长度，0 为不显示，1 为显示基本信息，2 为显示详细信息。

1. 在混合平面模型中使用无反射边界条件

如果计划在计算中同时采用无反射边界条件和混合平面模型，则首先要将混合平面定义为压强

2. 在并行版 FLUENT 中使用无反射边界条件

在无反射边界条件与并行求解器共同使用时，采用无反射边界条件的网格单元必须处于同一个分区中。为保证所有单元在同一个分区中，可以用人工方式进行网格分区。

8.24 用户自定义风扇模型

用户自定义风扇模型可以更新描述风扇周期性变化特征的型函数文件，这些特征包括风扇的压强跃升以及风扇绕流的径向和周向速度分量。比如，在计算一个风扇前后的压强跃升时，用户可以将压强跃升定义为一个常数，或将压强跃升定义为风扇轴向速度的多项式函数，或者用一个型函数文件将压强跃升定义为坐标和轴向速度的函数。在计算过程中，压强跃升如果随流场变化而变化，则需要在计算过程中周期性地更新型函数文件，其更新过程是根据用户在输入过程中定义的条件进行的，使用本节介绍的用户自定义风扇模型就可以完成完成这类文件的更新操作。

使用这种模型需要先用 FORTRAN、C 等语言编制一个可以读入并写出 FLUENT 型函数文件的可执行文件。这个可执行文件在 FLUENT 计算过程中根据用户定义的条件被调用和执行。

8.24.1 使用用户自定义风扇模型的步骤

创建用户自定义模型的步骤如下：

（1）在计算模型中将一个或多个区域标记为风扇区域，即在 Boundary Conditions（边界条件）面板中，选择风扇所在网格分区，并将此分区定义为风扇边界。

（2）在 User-Defined Fan Modal（用户自定义风扇模型）面板上输入预先制作完毕的可执行文件的名字，同时输入控制型函数文件读入、输出过程的指令。启动 User-Defined Fan Modal（用户自定义风扇模型）面板的菜单操作为：

Define->User-Defined->Fan Modal...

（3）初始化流场和型函数文件。

（4）在 Fan（风扇）面板中输入风扇参数。

（5）开始计算。

8.24.2 用户自定义风扇的例子

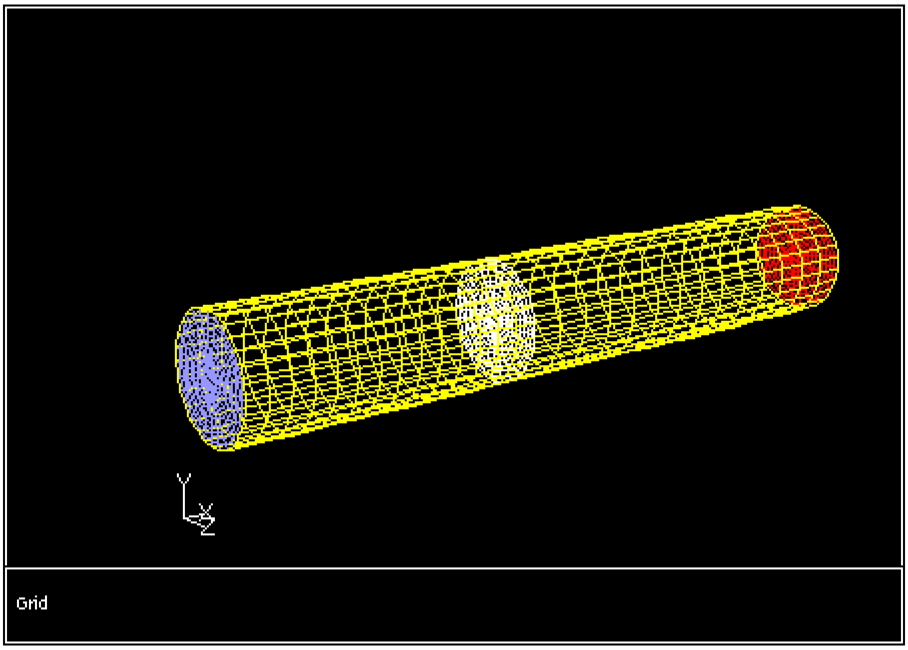


图 8-31 用户自定义风扇模型算例

看一个算例：如图 8-31 所示，圆柱区域入口空气速度为 10m/s，圆柱长 1.25m，直径 0.2m，被对称边界包围。在流场中部是一个圆形风扇，出口处使用压强出口条件。

采用用户定义风扇模型计算这个问题时，每隔一段时间 FLUENT 将输出型函数文件。这些变量代表风扇各环面上的变量平均值。环面尺寸取决于风扇大小和径向点的数量。

在型函数文件被输出时，FLUENT 将调用一个可执行文件执行下列任务：

（1）读入包含当前风扇上的流动条件的型函数文件。

（2）计算压强跃升、径向速度和旋转速度的值。

（3）输出一个包含上述计算结果的型函数文件。

FLUENT 随后读入新的型函数文件并继续进行计算。

1. 设定用户自定义风扇参数

用户自定义风扇模型参数在 User-Defined Fan Modal（用户自定义风扇模型）面板中输入，见图 8-32。

在风扇区域（Fan Zones）中可以选择用户自定义模型的应用区域，在本例中只有 fan-8 一个区域。FLUENT 中允许同时读入多个区域。

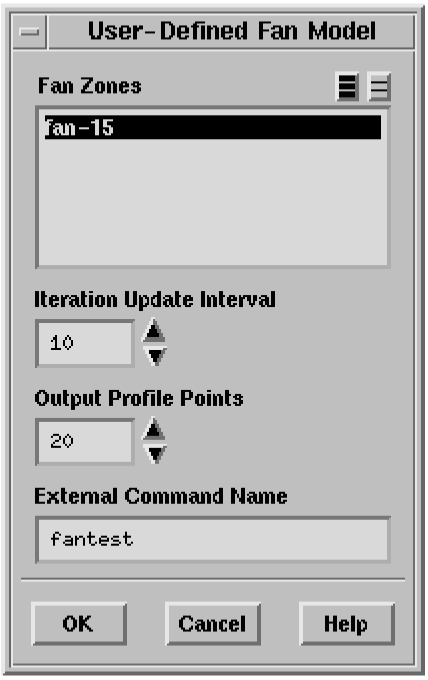


图 8-32 用户自定义风扇模型面板

可执行文件被调入后将每隔一段时间对型函数文件进行一次更新，更新的间隔在 Iteration Update Interval（迭代更新间隔）中输入。缺省设置的 10 表示每迭代 10 次对型函数文件进行一次更新。

型函数文件中的点数在 Output Profile Points（输出型函数点）中输入。型函数中的点数与外部可执行文件中的点数可以相同也可以不同。

最后在 External Command Name（外部命令名称）中输入可执行文件名。在当前算例中，可执行文件名为 fantest。

2. 用户自定义风扇程序的例子

可知行文件是从下面的 FORTRAN 程序，fantest.f，建立的。

```
c
c     This program is invoked at intervals by FLUENT to
c     read a profile-format file that contains radially
c     averaged data at a fan face, compute new pressure-jump
c     and swirl-velocity components, and write a new profile
c     file that will subsequently be read by FLUENT to
c     update the fan conditions.
c
c     Usage: fantest < input_profile > output_profile
c

      integer npmax
      parameter (npmax = 900)
      integer inp         ! input: number of profile points
      integer iptype      ! input: profile type (0=radial, 1=point)
      real ir(npmax)      ! input: radial positions
      real ip(npmax)      ! input: pressure
      real idp(npmax)     ! input: pressure-jump
      real iva(npmax)     ! input: axial velocity
      real ivr(npmax)     ! input: radial velocity
      real ivt(npmax)     ! input: tangential velocity
      character*80 zoneid
      integer rfanprof  ! function to read a profile file
      integer status
c
      status = rfanprof(npmax,zoneid,iptype,
     $     inp,ir,ip,idp,iva,ivr,ivt)
      if (status.ne.0) then
         write(*,*) 'error reading input profile file'
      else
         do 10 i = 1, inp
```

```
            idp(i) = 200.0 - 10.0*iva(i)
            ivt(i) = 20.0*ir(i)
            ivr(i) = 0.0
 10       continue
          call wfanprof(6,zoneid,iptype,inp,ir,idp,ivr,ivt)
       endif
       stop
       end
```

这个型函数文件将根据用户输入的 ptype 参数决定文件格式，文件中各种分布函数的名字是任意的，子函数 wfanprof 中给定的名字将在文件初始化后出现在 Fan（风扇）面板中作为型函数文件的名字。

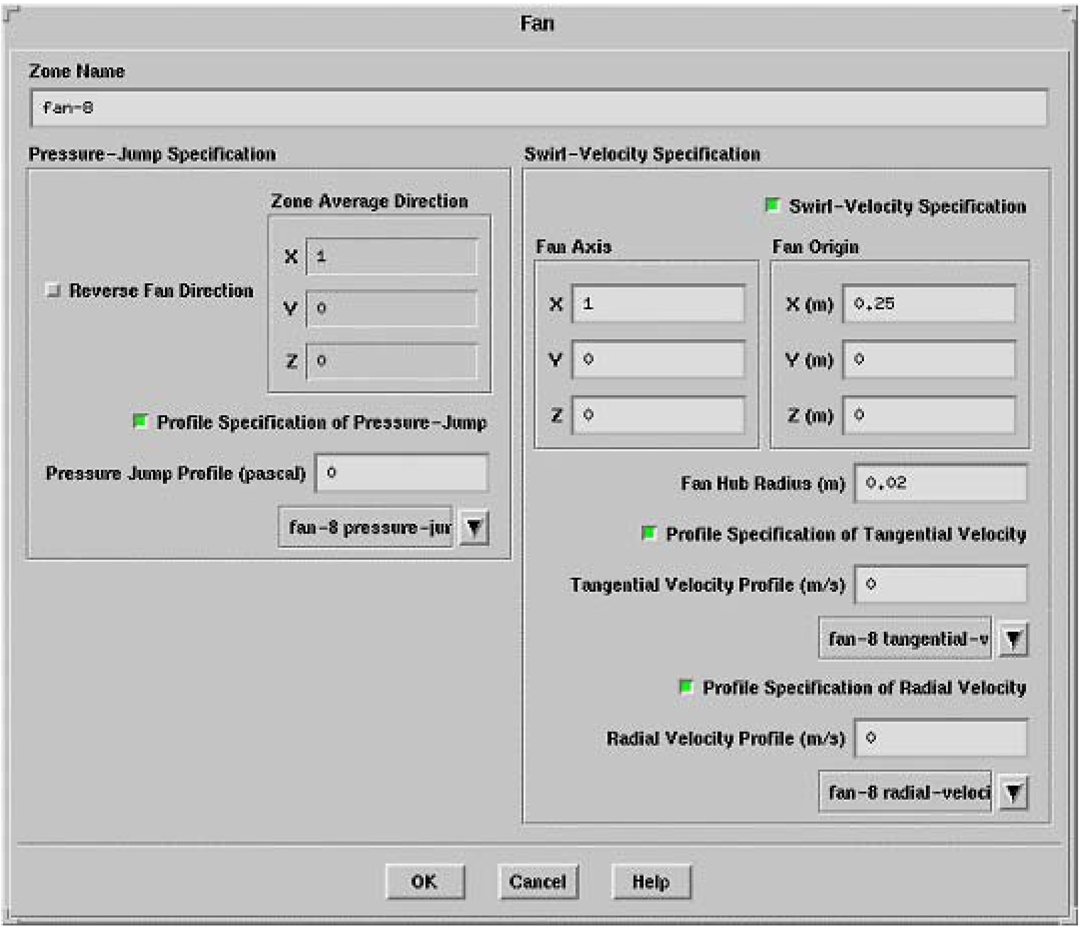


图 8-33 Fan（风扇）面板

3. 初始化流场和型函数文件

用户自定义风扇函数设置的下一步是初始化型函数文件。在 Solution Initialization（求解初始化）面板中，假设使用的是速度入口边界条件，在控制台窗口中键入命令 (update-user-fans)[1]，就可以创建型函数文件了。

4. 选择型函数

型函数文件生成后，在 Fan（风扇）面板中完成问题的设置。在这一步需要输入的参数为 Fan Axis（风扇轴）、Fan Origin（风扇原点）和 Fan Hub Radius（风扇桨毂半径），同时为计算压强跃升、切向速度和径向速度选择型函数。

5. 进行计算

在计算开始收敛时，控制台窗口中报告型函数文件每隔 10 个迭代步被更新一次。文件 fan-8-out.prof 由 FLUENT 中的可执行文件 fantest 执行读写操作。这个文件中的数据包括压强、压强跃升、轴向速度、径向速度、周向速度等变量的值。

6. 计算结果

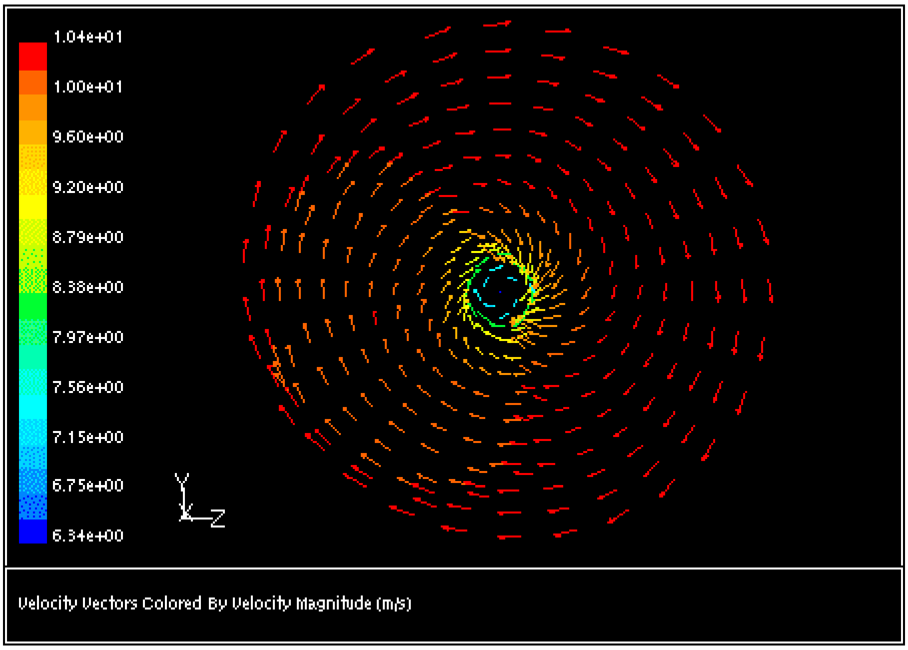


图 8-34 计算结果

[1] 括号是这个命令的一部分，不能删去不用。

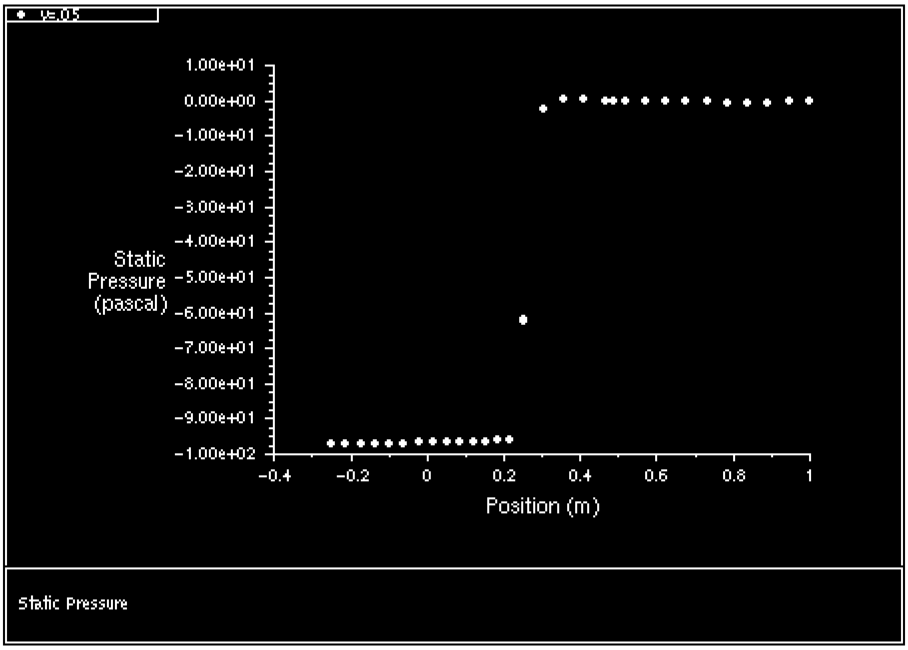


图 8-35 风扇前后压强的跃升

计算结果如图，如同计算前设想的一样，在风扇上没有径向速度分量，同时周向速度分量随半径增加而增加。

最后检查一下压强跃升。使用 XY 图形可以将静压显示为 x 的分布函数。图 8-35 的位置在 y=0.05m，大约 0.5 倍半径的地方。由输入条件可知，压强跃升大约为 97.8Pa/m，从图 8-35 中可以对此得到验证。

8.25 换热器模型

换热器由大量管道组成，在流场计算中很难计算大量管道和散热片组成的复杂几何构型，考虑到换热器的影响是在流场中造成压强降，同时将热量传递给另一种流体，所以 FLUENT 中用参数集合的形式建立换热器的数学模型。FLUENT 中的换热器模型有两种，一种是简化效率模型，另一种是传热单元数量（number-of-transfer-units，NTU）模型。上述模型可以计算排热量固定时冷却剂的入口温度，或者已知冷却剂入口温度时计算总的排热量。对于简化效率模型，冷却剂可以是单相的或两相的。

8.25.1 换热器模型概述与限制条件

1. 概述

换热器的一个典型例子是冷却器。在冷却器中，冷却剂流过管道，将管道外面气体中的热量带走，从而达到降低气体温度的目的。

在冷却器中，冷却剂的温度是沿流动方向变化的，因此可以将冷却器假定为沿流动方向排列的一系列冷却单元的集合。在计算过程中可以先计算出冷却剂在每个冷却单元入口处的温度，再算出每个单元的热量损耗。

采用这种物理模型进行计算时，首先要将冷却器所在空间区域定义为一个或几个流体区域，然后再定义冷却剂的路径、冷却单元的数量、冷却剂属性和操作条件（包括压强降参数、换热器效率、冷却剂流量等等）。

热交换模型假定换热器是排列紧密的，也就是说换热器外部气流方向是均匀不变，而冷却剂则是在大量相互平行的管道中流动。冷却剂的流动途径可以是曲折的，并在往返过程中形成数个通道。可以人为地定义通道的主要方向，以及从一个通道向另一个通道转折的方向，同时还可以定义外部气流的方向。

2. 限制条件

热交换模型的限制条件如下：

（1）换热器的形状大致为矩形。

（2）气流方向必须与由矩形定义的某个坐标轴平行。

（3）在压力损失系数的计算中忽略了加速效应的影响。

（4）在简化效率模型中，气体热容变化小于冷却剂的热容变化。

（5）在 NTU 模型中不考虑冷却剂的相变。

8.25.2 换热器模型理论

在 FLUENT 中，换热器被定义为有动量和热量交换的流体区域。压强损失用动量方程中的动量汇来表示，热交换则用能量方程中的热源项表示。

NTU 模型是 FLUENT 缺省设置的换热器模型，其特征如下：

（1）同时计算气体和冷却剂的热容，并用较小的一个进行热交换计算。

（2）可以计算气体与冷却剂之间的热交换过程。

（3）可以计算气体回流情况下的热交换。

（4）可以用于密度变化气体的计算。

（5）适用于串行版和并行版的 FLUENT 求解器。

简化效率模型的特征为：

（1）可以计算热量从冷却剂流向气体的热交换过程。

（2）冷却剂性质可以被定义为压强和温度的函数，因此可以在计算中考虑冷却剂相变的影响。

（3）适用于串行版和并行版的 FLUENT 求解器。

（4）可以计算由换热器组构成的换热器网络。

1. 流向压强降

压强降用 FLUENT 中的多孔介质模型定义。流向压强降可以定义为：

$$\Delta p = \frac{1}{2} f \rho_m U_{A\min}^2 \qquad (8\text{-}78)$$

式中 Δp 为流向压强降，f 为流向压强降损失系数，ρ 为平均气体密度，$U_{A\min}$ 为流管面积最小处气体的速度。

压强损失系数的计算公式为：

$$f = \left(K_c + 1 - \sigma^2\right) - \left(1 - \sigma^2 - K_e\right)\frac{v_e}{v_i} + 2\left(\frac{v_e}{v_i} - 1\right) + f_c \frac{A}{A_c}\frac{v_m}{v_i} \qquad (8\text{-}79)$$

式中 σ 为最小流量密度，K_c 为进口损失系数，K_e 为出口损失系数，A 为气体一侧的表面积，A_c 为最小流管面积，f_c 为换热器摩擦系数，v_e 出口处的比容，v_i 为进口处的比容，$v_m = \frac{1}{2}\left(v_e + v_i\right)$ 为平均比容。其中换热器的摩擦因子定义为：

$$f_c = a \,\mathrm{Re}_{\min}^b \qquad (8\text{-}80)$$

式中 a 为换热器摩擦系数，b 为摩擦指数，$\mathrm{Re}_{\min}$ 为用最小流管面积处速度计算的雷诺数，其定义为：

$$\mathrm{Re}_{\min} = \frac{\rho_m U_{A\min} D_h}{\mu_m} \qquad (8\text{-}81)$$

式中 μ_m 为气体平均粘度，D_h 为水力直径，其定义为：

$$D_h = 4L\left(\frac{A_c}{A}\right) \qquad (8\text{-}82)$$

$U_{A\min}$ 可以用下式计算：

$$U_{A\min}=\frac{U}{\sigma} \qquad (8\text{-}83)$$

式中U为气体速度。

2. 热交换效率

在简单效率模型中，热交换效率ε被定义为实际热流与最大热流之比。最大热流计算公式为：

$$q_{\max}=C_{\min}\left(T_{in,hot}-T_{in,cold}\right) \qquad (8\text{-}84)$$

$T_{in,hot}$和$T_{in,cold}$为入口处冷、热流体的温度，$C_{\min}$的计算公式为：

$$C_{\min}=\min\left[\left(\dot{m}c_p\right)_{hot},\left(\dot{m}c_p\right)_{cold}\right] \qquad (8\text{-}85)$$

因此实际热流量为：

$$q=\varepsilon C_{\min}\left(T_{in,hot}-T_{in,cold}\right) \qquad (8\text{-}86)$$

其中的效率取决于换热器的几何形状及流动类型（平行流、反向流、横向流等等）。虽然是作为换热器的总体性质定义的，它也可以用于定义局部计算。

在 NTU 模型中，FLUENT 用热容比和交换器数量计算效率：

$$\varepsilon=1-\exp\left[-\frac{1}{C_r}N_{tu}^{0.22}\left(1-e^{-C_r N_{tu}^{0.78}}\right)\right] \qquad (8\text{-}87)$$

C_r为$C_{\min}$与$C_{\max}$之比。N_{tu}应该根据流量大小定义。这里的N_{tu}是根据冷却器单元面积与总面积的比例进行了缩尺处理的。

对于每个冷却器单元来说，气体的入口温度是用边界上气体温度的质量平均值计算得到的。这样自动将边界上的气体回流计入计算中。

3. 热耗散

热耗散是针对冷却单元中的每个网格进行计算的。从冷却剂到气体的热耗散可以是正值，也可以是负值。

在简单效率模型中，网格上的热交换量是换热器效率、温差和热容流量（流量与比热的乘积）的函数。冷却单元的热流量则是全部网格热交换量的和。

在 NTU 模型中，热交换量用下式计算：

$$q=\varepsilon C_{\min}\left(T_{in,coolant}-T_{in,gas}\right) \qquad (8\text{-}88)$$

$T_{in,coolant}$ 为冷却剂进口温度，$T_{in,gas}$ 为气体进口温度。每个网格上的热交换量则与其占有的体积比例成正比。

无论使用哪个模型，换热器的热交换量等于各个单元热交换量之和。

4. 换热器组的连通性

如果在计算中将换热器分解为多个热交换单元，则各单元的冷却剂流量是根据各自的体积在换热器总体积中所占比例来计算的。而各单元上的焓则是根据质量流比例来计算的。

6.25.3 使用换热器模型

设置换热器模型的步骤如下：

（1）在 Energy（能量）面板中加入能量计算。

（2）在 Heat Exchanger（换热器）面板中给定换热器模型的输入参数，如图 8-36 所示。启动 Heat Exchanger（换热器）面板的菜单操作为：

Define->User-Defined->Heat Exchanger...

- 在 Fluid Zone（流体区域）中，选定换热器所占据的区域。
- 定义冷却剂入口方向（Coolant Inlet Direction）和通道转换方向（Pass-to-Pass Direction）。
- 定义换热器阵列，给定通道数量（Number of Passes）、通道排数（Number of Rows/Pass）和通道列数（Number of Cols/Pass）。
- 指定用于计算的换热器模型，即设定使用 NTU 模型，还是简化效率模型。
- 给定冷却剂性质和条件（冷却剂流量、入口温度和冷却剂比热）。
- 在 Heat Exchanger Core Modal（换热器模型）列表中定义压强降参数和换热器效率。
- 点击 Apply（应用）按钮保存所有设置。
- 如果有其他换热器，则重复进行上面的步骤。

下面详细介绍相关的操作：

1. 为换热器选择区域

在 Fluid Zone（流体区域）列表中选定相应的区域名称即可。

2. 定义冷却剂入口和通道转换方向（Pass-to-Pass Directions）

在已知冷却剂方向和流动路径时，可以在 Coolant Inlet Direction（冷却剂入口方向）和 Pass-to-Pass Directions（通道转换方向）中，相对于换热器单元的方向，定义这些方向。如果事先无法确定这些方向，则可以用面工具定义这些方向。

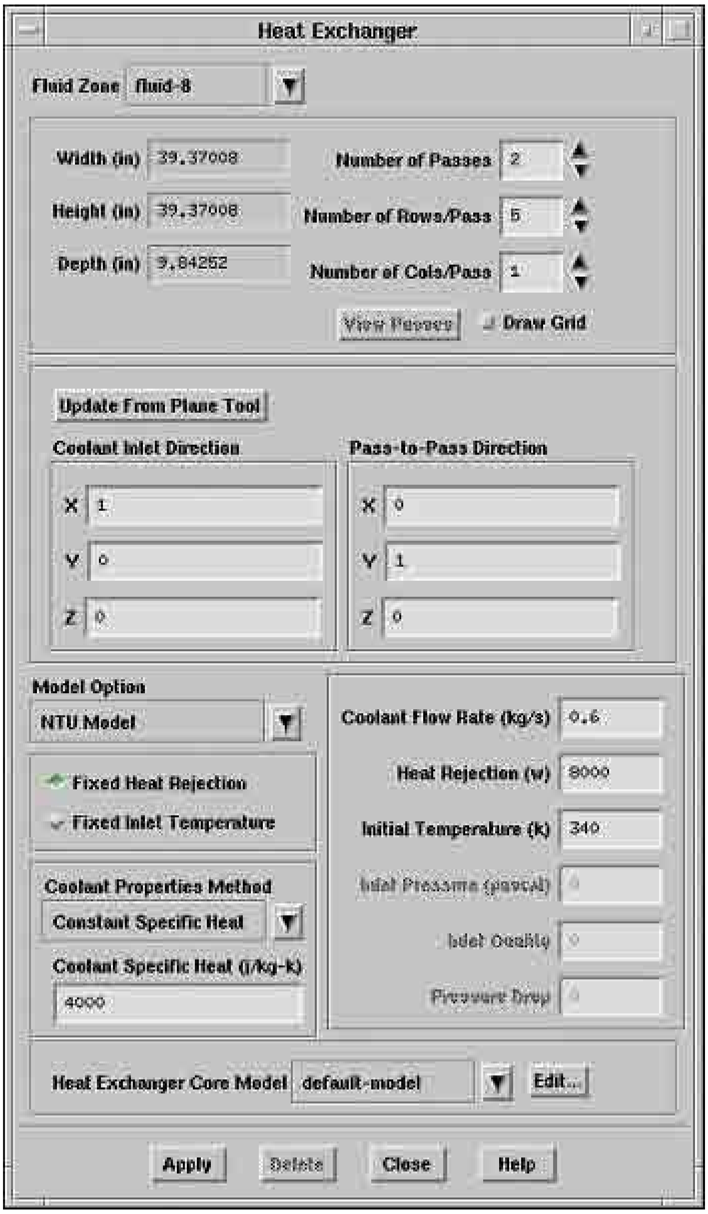


图 8-36 Heat Exchanger（换热器）面板

3. 定义换热器单元

如 8.25.1 小节中所说，代表换热器的流场区域在计算中由多个换热器单元构成。换热器单元则用指定的通道数量、每个通道的换热器单元排数、列数及冷却剂入口和通道转换方向等参数定义的。换热器单元编号从 0 开始，到 n-1 为止，n 是换热器单元的总数。

在 Heat Exchanger（换热器）面板中给定 Number of Passes（通道数）、Number of Rows/Pass（每个通道上的单元排数量）和 Number of Cols/Pass（每个通道上的单元列数量）。

输入完成后，点击 Apply（应用）按钮，然后点击 View Passes 显示所定义的单元划分情况。

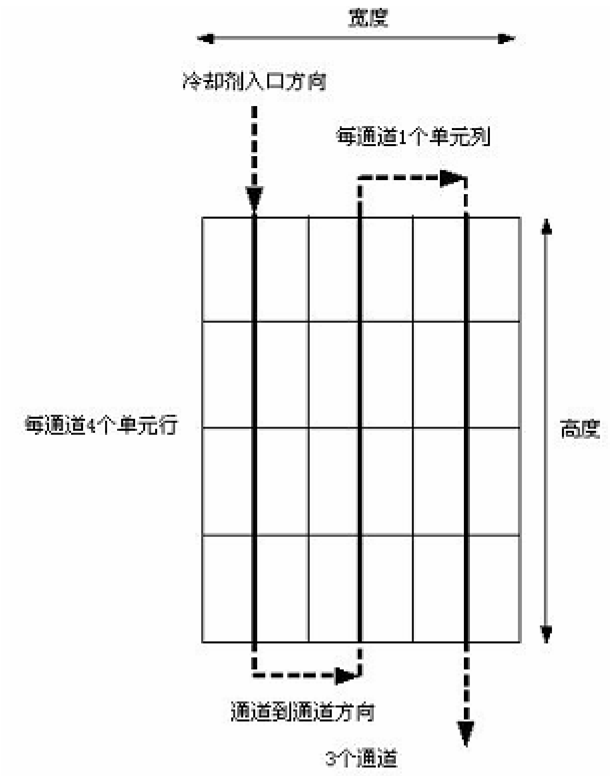


图 8-37 换热器单元（1 x 4 x 3 个单元）

4. 选择换热器模型

模型选择在 Modal Option（模型选项）列表中完成，可以选择 NTU Modal（NTU 模型），或 Simple Effectiveness Modal（简化效率模型）。

5. 定义冷却剂性质和条件

冷却剂性质可以在 Coolant Properties Method（冷却剂性质定义方法）列表中给定。需要在这里定义 Coolant Flow Rate（冷却剂流量）、Constant Specific Heat（常值定压比热）等等。还可以计算固定散热量时的冷却剂进口温度，或在进口温度已知时计算总的散热量。

6. 设定压强降参数和效率

在控制台窗口中键入(set! Auto-set-porous? #t)命令可以启动换热器模型定义。有三种方法可以定义这个模型中的参数：

（1）采用模型中的缺省值。

（2）输入新的参数值，改变缺省模型定义。

在换热器模型面板中，先在 Name（名称）栏中键入新模型的名称，然后在 Gas-Side Pressure Drop（气体一侧的压强降）参数组中输入方程（8-79）中的各个参数：

- 最小流量密度（σ）。
- 进口损失系数（K_c）
- 出口损失系数（K_e）
- 气体一侧表面积（A）
- 最小流管面积（A_c）
- 换热器摩擦系数（a）
- 换热器摩擦指数（b）

在 Effectiveness（效率）组下，根据选择的模型（NTU 模型或简化效率模型）定义模型计算的点数和效率的值。最后点击 Change/Create（修改/创建）按钮将新创建的模型加入效率模型库。

（3）读入外部定义文件。

也可以从外部文件中读入上述定义参数。下面是一个例子：

```
("modelname"
  (0.73 0.43 0.053 5.2 0.33 9.1 0.66)
  ((1 1.0 .6234) (2 2.0 0.5014) (3 3.5 0.3932)
   (4 5.0 0.3244) (5 6.5 0.2762) (6 8.0 0.2405)
   (7 10.0 0.2050) (8 12.0 0.1785) (9 15.0 0.1495)))
```

文件的第一行是模型名称，第二行是压强降参数（σ、K_c、K_e、A、A_c、a、b），第三行是定义效率型函数的点（简化效率模型），或质量流量与 N_{tu} 型函数（NTU 模型），每个点上的参数格式如下：

简化效率模型：

（点　速度　效率）

NTU 模型：

（点　质量流量　N_{tu}）

样例文件中定义了 9 个点。

外部文件可以用下列步骤读入：

（1）在 Heat Exchanger Modal（换热器模型）面板中，点击 Read...（读入）按钮。

（2）在文件选择对话框中，给定 HXC Parameters File（HXC 参数文件）名，然后点击 OK 按钮。

8.25.4 使用换热器组

如果换热器跨越多个流体区域，或者要将多个换热器中的冷却剂流动路径连接起来，可以使用换热器组。步骤如下：

（1）在 Energy（能量）面板中加入能量计算。

（2）在 Heat Exchanger Group（换热器组）面板中进行参数定义，启动这个面板的菜单操作为：

Define->User-Defined->Heat Exchanger Group...

1）在 Name（名称）栏中，输入换热器组的名称。

2）在 Fluid Zones（流体区域）中，选择换热器占用的区域。

3）点击 Modal（模型）标签。

- 确定计算用模型（NTU 模型或简化效率模型）。
- 在 Connectivity（连通性）下，如果存在连通关系，则选择 Upstream（上游）换热器组。
- 在 Gas Flow Direction（气流方向）下，定义气流方向参数。
- 在换热器模型列表中，定义压强降参数和效率。

4）点击 Geometry（几何）标签，在这里定义换热器单元和冷却剂入口方向和通道转换方向。

5）点击 Coolant（冷却剂）标签，定义冷却剂性质和条件，如比热、流量和初始温度。

6）如果存在辅助冷却剂，点击 Auxiliary Stream（辅助流）标签。定义辅助冷却剂的流量、温度和品质等参数。

7）点击 Create（创建）按钮创建模型，或 Replace（更换）按钮替代原来模型。

（3）如果换热器组由多个流体区域组成，而用户需要覆盖此前在这些区域上的参数设置的话，可以点击 Set...（设置）按钮打开 Heat Exchanger（换热器）面板，重新进行设置并覆盖过去的参数即可。可以被覆盖的参数包括单元阵列参数、冷却剂流动路径方向矢量和换热器模型。

1. 为换热器组选择流体区域

在 Fluid Zones（流体区域）列表中选择将被定义为换热器组的区域。在这些区域中的冷却剂流动应该是互相平行的。注意同一个区域不能被换热器组重复引用。

2. 选择上游换热器组

如果需要将当前的组与其他组相互连通，可以选择上游换热器组。每个组最多只能有一个上游组和一个下游组。同时一个组不能与它自身相连通。在 Heat Exchanger Group（换热器组）面板中，Modal（模型）标签下的 Connectivity（连通性）栏目中的 Upstream（上游）列表中选择相应的换热器组。

需要注意的是连通性是在创建换热器组时定义的。这种连通关系是不能被改变的，除非删除带有连通性的组，并创建一个新的带有新的连通关系的组。

3. 定义冷却剂入口方向和通道转换方向

冷却剂比热可以定义为常值比热（Constant Specific Heat），或者用 UDF 函数计算其焓，函数应该返回下列值：

（1）计算温度、压强、品质所需的焓。

（2）计算焓和压强所需的温度。

（3）计算比热所用的温度和压强。

UDF 函数的类型定义为：

DEFINE_SOURCE(udf_name, cell_t c, Thread *t, real d[n], int index)

其中参数 n 取值的含义为，0 为温度，1 为压强，2 为品质。变量 index 取 0 时为焓，1 为温度，2 为比热。UDF 函数应该返回：

real value; （说明：根据 index 取值不同返回的温度、焓或定压比热的值）

4. 定义辅助冷却剂流

可以在换热器中加入或去掉辅助冷却剂流。需要注意的是，辅助冷却剂不能用于单一区域，冷却剂的流量必须小于冷却剂主流流量，辅助冷却剂的性质与主冷却剂相同。给定质量流量、温度和品质就可以完成辅助冷却剂的定义。

5. 初始化冷却剂温度

在换热器组之间具有连通性时，所有冷却剂的入口温度都等于其上游换热器组的入口温度，因此最上游的入口温度会在缺省情况下传播到下游所有组中。如果希望改变缺省设置，可以对每个组单独进行设置，但是每次改变的值都将传播到下游的组中。如果希望每个组的初始温度是不同的，则必须从第一个组开始顺序改变温度的值。在组内所有换热器必须使用固定入口温度选项，并拥有相同的入口温度值，也就是说同一组内的换热器不能设置不同的入口温度。

8.25.5 换热器模型的后处理

可以观看总的散热量、出口温度、入口温度等参数的计算结果，其文本命令为：

define->user-defined->heat-exchanger->heat-exchanger-report

在屏幕提示下，可以通过定义 id/name 参数，选择报告特定区域上的计算结果。如果需要查看换热器组的连通性，可以使用命令：

(report-connectivity)

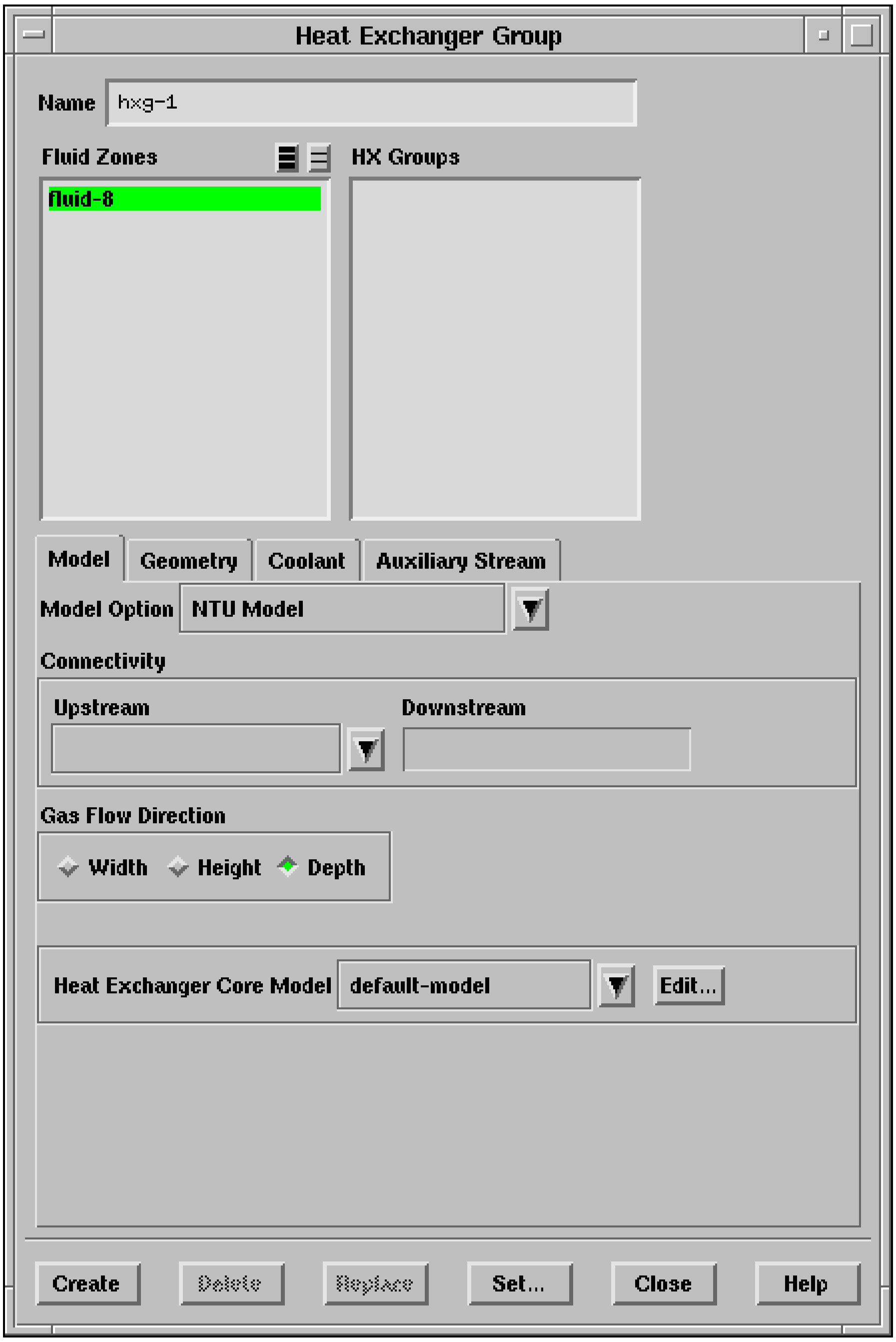


图 8-38 换热器组面板

6.26 边界型函数

边界型函数可以对指定变量进行灵活的定义，其数据来源是试验数据、其他程序的计

算结果或在 Write Profile（写型函数）面板中生成的解。

6.26.1 边界型函数的类型

边界型函数有四种类型：

（1）点型函数（point profiles），即用无序排列的点进行定义的型函数。这类型函数可以用 Write Profiles（写型函数）面板生成。FLUENT 用最靠近边界的点定义边界条件，因此无序排列的点应该具有一定的密度，以保证边界条件的精度。

（2）线型函数（line profiles），即二维问题中用有序点集进行定义的型函数。比如在外部程序中输出的某个变量的分布函数，在 FLUENT 中可以被用作该函数在边界上的边界条件。

（3）网格型函数（mesh profiles），即三维问题中用带两个下标的有序点集定义的型函数。比如在用结构化网格计算出的某个变量的分布，在 FLUENT 中可以被用作该函数在三维边界上的边界条件。

（4）径向型函数（radial profiles），即用径向有序点集定义的型函数。这类型函数适用于变量是且仅是半径的函数的情况。

8.26.2 边界型函数的文件格式

型函数文件的格式比较简单。这类文件可以包含任意个变量的型函数。每个型由一个定义型的名称、类型（点、线、网格、径向）的头、定义点数量及后面任意个变量组成。这些变量中的一部分是坐标点，其余部分是边界条件数据。

因为 FLUENT 在读入型函数时不做单位转换，所以所有数据必须采用国际单位制。数据之间可以用空格、tab 键和新的行来进行分隔。下面是型函数文件的语法格式：

```
((profile1-name point|line|radial n)
 (field1-name a1 a2 ... an)
 (field2-name b1 b2 ... bn)
 .
 .
 .
 (fieldf-name f1 f2 ... fn))

((profile2-name mesh m n)
 (field1-name   a11 a12 ... a1n
                a21 a22 ... a2n
                .
                .
```

```
                  .
                  am1 am2 ... amn)
 .
 .
 .
 (fieldf-name     f11 f12 ... f1n
                  f21 f22 ... f2n
                  .
                  .
                  .
                  fm1 fm2 ... fmn))
```

下面是一个在入口边界上定义速度和湍流动能 k 和湍流耗散率 ε 的型函数：

```
((turb-prof point 8)
(x
   4.00000E+00   4.00000E+00   4.00000E+00   4.00000E+00
   4.00000E+00   4.00000E+00   4.00000E+00   4.00000E+00 )
(y
   1.06443E-03   3.19485E-03   5.33020E-03   7.47418E-03
   2.90494E-01   3.31222E-01   3.84519E-01   4.57471E-01 )
(u
   5.47866E+00   6.59870E+00   7.05731E+00   7.40079E+00
   1.01674E+01   1.01656E+01   1.01637E+01   1.01616E+01 )
(tke
   4.93228E-01   6.19247E-01   5.32680E-01   4.93642E-01
   6.89414E-03   6.89666E-03   6.90015E-03   6.90478E-03 )
(eps
   1.27713E+02   6.04399E+01   3.31187E+01   2.21535E+01
   9.78365E-03   9.79056E-03   9.80001E-03   9.81265E-03 )
)
```

8.26.3 使用边界型函数

在计算中使用边界型函数的步骤如下：

（1）创建符合格式要求的边界型函数文件。

（2）读入边界型函数的方法有二：

1）Define->Profiles...

2）File->Read->Profile...

（3）在相应的边界条件面板中，比如在压强入口条件和速度入口条件中，从列表中选择相应的型函数即可。

每个型函数文件中都包含一个或几个变量的型函数，在型函数文件被载入 FLUENT 后，可以检查变量数量和名称，也可以删除型函数，操作界面为边界型函数（Boundary Profiles）面板，启动这个面板的方法为：

Define->Profiles...

8.26.4 边界型函数的重定向

在三维计算中，FLUENT 可以将边界型函数进行重定向，即改变原来边界型函数的方向。这样就可以将原来定义在某个方向的型函数应用于任意方向，从而增加了型函数使用的灵活性。比如可以将原来定义在 X 方向的边界条件，重定向后应用于 Z 方向。

重定向的操作步骤为：

（1）定义并载入边界型函数文件。

（2）在 Boundary Profiles（边界型函数）面板中，选择需要重新定向的型函数，然后点击 Orient...（定向）按钮，打开的 Orient Profile（型函数定向）面板。

（3）在名称栏中输入新的文件名。

（4）在 New Fields（新变量场）栏中给定变量数量，该数量应该等于变量数加 1。

（5）定义坐标系（x,y,z），即从 Compute From...（计算起始位置）下面的列表中选择适当的局部坐标系。

（6）定义新的型函数中的矢量场名称及其对应的坐标轴方向。

（7）定义新的型函数中的标量场，注意需要开启 Treat as Scalar Quantity（作为标量处理）选项。

（8）在 Orient To...（重定向到……）中定义局部坐标系的原点及坐标方向。除了直接指定坐标方向外，还可以用面工具进行定义。

（9）点击 Create（创建）按钮创建新的型函数文件。

8.26.5 用边界型函数定义瞬态边界条件

标准瞬态型函数文件的格式为：

```
((profile-name transient n periodic?)
(field_name-1 a1 a2 a3 .... an)
(field_name-2 b1 b2 b3 .... bn)
.
.
.
(field_name-r r1 r2 r3 .... rn))
```

其中 profile-name 为文件名，transient 为瞬态文件标志，n 为每个变量的个数。periodic?用于说明所定义的边界条件是否为周期性边界条件，这个值等于 1 则为周期性边界条件[2]，等于 0 则为非周期边界条件。field_name 为变量名，后面为变量的具体数值，其中必须有一个变量为 time（时间），时间变量必须按升序排列。下面是一个瞬态型函数文件的例子：

```
((sampleprofile transient 3 1)
(time
1
2
3
)
(u
10
20
30
)
)
```

其中定义了两个变量，第一个是时间变量 time，第二个是速度变量 u，每个变量赋 3 个值，并且为周期性边界条件。

8.26.6 用边界型函数定义湍流边界条件

在 FLUENT 中可以使用型函数文件将试验数据或经验公式引入计算，从而达到精确定义湍流边界层的目的。在 Boundary Conditions（边界条件）面板中，选择需要定义湍流参数的入口、出口或远场边界，然后点击 set（设置）按钮进入相应的边界条件设置面板。在面板的下部通常就是湍流参数的定义栏。在创建了相应的型函数，并将其读入算例后，进入这个面板，然后根据选定的模型在 Turbulence Specification Method（湍流定义方法）下拉菜单中选择相应的定义办法，再在这个下拉菜单下方的参数输入栏右面选择相应的型函数文件就可以了。不同的湍流模型需要使用的湍流定义方法分别为：

（1）使用 Spalart-Allmaras 模型时，相应选择 Turbulence Viscosity（湍流粘度）或 Turbulence Viscosity Ratio（湍流粘度比），再选择各参数所对应的型函数名，FLUENT 就可以将 μ_t / μ 与密度、分子粘度组合在一起，计算出湍流粘度 $\tilde{\upsilon}$ 在边界上的值。

（2）使用 $k-\varepsilon$ 模型时，相应地选择 K and Epsilon 选项，然后在 Turb. Kinetic Energy（湍流动能）和 Turb. Dissipation Rate（湍流耗散率）旁选择相应的型函数即可。

（3）使用 $k-\omega$ 模型时，相应地选择 K and Omega，然后在 Turb. Kinetic Energy（湍流动能）和 Turb. Dissipation Rate（湍流耗散率）旁选择相应的型函数即可。

[2] 即边界条件随时间呈周期性变化。

（4）使用雷诺应力模型（RSM）时，应该在 Turbulence Specification Method（湍流定义方法）下拉菜单中首先选择 K and Epsilon，并在 Turb. Kinetic Energy（湍流动能）和 Turb. Dissipation Rate（湍流耗散率）旁选择相应的型函数。在自动出现的 Reynolds-stress Specification Method（雷诺应力定义方法）下拉菜单中选择 Reynolds-stress components（雷诺应力分量），并在相应的雷诺应力分量旁选择相应的型函数。

8.27 将变量的值设为固定值

FLUENT 可以将一个或多个变量的值，在某个区域中设定为固定值。需要注意的是，在分离算法中，只有速度分量、温度和组元浓度可以被固定。

8.27.1 概述

如果不必考虑某个变量的变化过程和细节，就可以用固定其函数值的做法进行流场计算。比如在计算推进器流场时，如果不需要考虑螺旋桨前后的速度场细节，就可以不必花费时间和系统资源对它进行计算，而只需将螺旋桨出口处的速度设为固定值，作为一个边界条件开始计算流场其余部分的流动。

可以被固定的变量包括速度分量、湍流变量、温度、焓、组元质量浓度、用户自定义变量等。

8.27.2 在区域中固定函数值的操作流程

在一个网格区域内固定一个或几个变量的步骤如下：

（1）在 Fluid（流体）面板或 Solid（固体）面板中，打开 Fixed Values（固定值）选项。

（2）将被选变量设为固定值。

1. 固定速度分量

固定速度分量即将速度在各个坐标方向上的分量设为固定值。坐标可以是直角坐标，也可以是柱坐标。

2. 固定温度和焓

如果需要求解能量方程，则可以在网格区域上固定温度（Temperature）的值，温度的单位为 K。

如果计算中使用了非预混燃烧模型，则可以固定焓（Enthalpy）的值，焓的单位为

m^2/s^2。

3. 固定组元浓度

如果计算中不考虑化学反应，可以固定组元浓度的值。

4. 固定湍流变量

FLUENT 中使用的湍流模型包括 $k-\varepsilon$ 模型、$k-\omega$ 模型、Spalart-Allmaras 模型和雷诺应力模型等。

在 $k-\varepsilon$ 模型和 $k-\omega$ 模型中，可以固定湍流动能（Turbulence Kinetic Energy）和湍流耗散率（Turbulence Dissipation Rate）的值。

在 Spalart-Allmaras 模型中可以固定修正的湍流粘度（Modified Turbulent Viscosity）的值。

在雷诺应力模型中则除了可以固定湍流动能（Turbulence Kinetic Energy）和湍流耗散率（Turbulence Dissipation Rate）外，还可以固定雷诺应力项。

5. 固定用户自定义的变量

通过给定用户定义变量（User defined scalar-n）的值，可以将自定义变量的值设为固定值。

8.28 定义质量、动量、能量和其他源项

在计算区域中可以定义一些变量的源项，这些变量包括质量、动量、能量等等。源项通常被定义在一个网格单元，或一组网格单元上。在定义质量源项的同时需要定义一个与之相配合的动量源项，如有必要，还要定义相应的能量和湍流源项。

正的源项称为“源”，而负的源项则称为“汇”。所有源项均采用国际单位制。从物理角度上看，源项代表的是某个物理量的创生与消失。

例如，在一个水池中有一个入水口，水通过入水口不断流入水池。在这种情况下，我们可以将入水口定义为流场的入口边界，也可以把它看作一个“生成”质量的源。在实际计算中，通常是在不适合采用入口边界的情况下使用源项。

1. 定义源项的操作流程

定义源项的步骤如下：

（1）在 Fluid（流体）面板或 Solid（固体）面板中打开 Source Terms（源项）选项。

（2）对源项进行适当的设置。

2. 质量源项

质量源项的单位是 kg/m^3，在连续方程中用 S_m 表示。如果计算中需要考虑流体的组元成分，则在每个组元的连续方程中都可以用单独的源项 S_i 来表示组元的生成或消失。组元的源项与混合物的源项之间存在一个连续性关系，即组元的源项之和应该等于混合物的源项。

3. 动量源项

动量方程中的作用力项，从数学角度看，就是动量的源项，因此动量源项在动量方程中也是包含在作用力项中的，其单位为 N/m^3。定义动量源时，需要定义 X 方向动量（X Momentum）、Y 方向动量（Y Momentum）和 Z 方向动量（Z Momentum）。

4. 能量源项

热源的概念比较容易理解，任何释放热量的空间区域都可以被模型化为热源。将这个概念加以推广，任何释放能量的空间区域都可以被看作能量源。能量源的单位为 W/m^3，在能量方程中，能量源用 S_h 项表示。

5. 湍流源项

FLUENT 中使用的湍流模型包括 $k-\varepsilon$ 模型、$k-\omega$ 模型、Spalart-Allmaras 模型和雷诺应力模型等。

在 $k-\varepsilon$ 模型和 $k-\omega$ 模型中，需要定义湍流动能（Turbulence Kinetic Energy）和湍流耗散率（Turbulence Dissipation Rate）的源项。

在 Spalart-Allmaras 模型中需要定义修正的湍流粘度（Modified Turbulent Viscosity）的源项。

在雷诺应力模型中则除了定义湍流动能（Turbulence Kinetic Energy）和湍流耗散率（Turbulence Dissipation Rate）外，还要定义雷诺应力项。

其他可以定义的源项还包括燃烧模型中的平均混合物浓度和增量项、P-1 辐射模型中的辐射源项等等。

第 9 章 材料性质

本章重点介绍应用 Fluent 软件进行流体计算过程中，非常关键的一个步骤——流体材料的设定，包括物性参数的计算方程和详细的参数设定过程。本章也简要介绍关于多相流模型（VOF 模型、混合物模型、欧拉模型）等参数的设定，包括各模型中物质物性参数的设定。读者通过学习本章，可以掌握在 Fluent6.1 软件里调用数据库里已有的物性参数，和改变、添加自己的物性参数的方法和技巧。

9.1 物性参数设定简介

在建立数学模型中非常关键的一步便是正确设定所研究物质的物性参数。在 Fluent6.1 里，物性参数的设定是在 Materials（材料）面板中完成的。Materials（材料）面板如图 9-1。

在面板中需要设定的参数包括：

（1）密度和（或）分子量。

（2）粘度。

（3）比热。

（4）热传导系数。

（5）质量扩散系数。

（6）标准状态下的焓。

（7）分子动力论参数。

这些参数可以是温度或组成函数，而温度和组元的变化方程可以是多项式函数、阶梯函数或分段多项式函数。Fluent 设置流体物性的这个特点给计算带来很大方便，尤其是在温度场的变化非常复杂，物性参数很难用单个函数来表示的情况下尤其如此。单个组元物性参数的变化可以由用户指定或者由分子动力论确定。

Materials（材料）面板会显示已被激活的物质所有需要设定的物性参数。需要注意的是，如果用户定义的属性需要用能量方程来求解（例如用理想气体定律求密度，用温度函数求粘度），Fluent 软件会自动激活能量方程。在这种情况下必须要设定材料的热力学条件和其他相关参数。

对于固体材料来说，需要定义材料的密度、热传导系数和比热。如果模拟半透明物质，还需要设定物质的辐射属性。固体物质热传导系数的设置很灵活，既可以是常数值，也可

以是随温度变化的函数，甚至由用户自定义函数（UDF，User Defined Function）来定义。

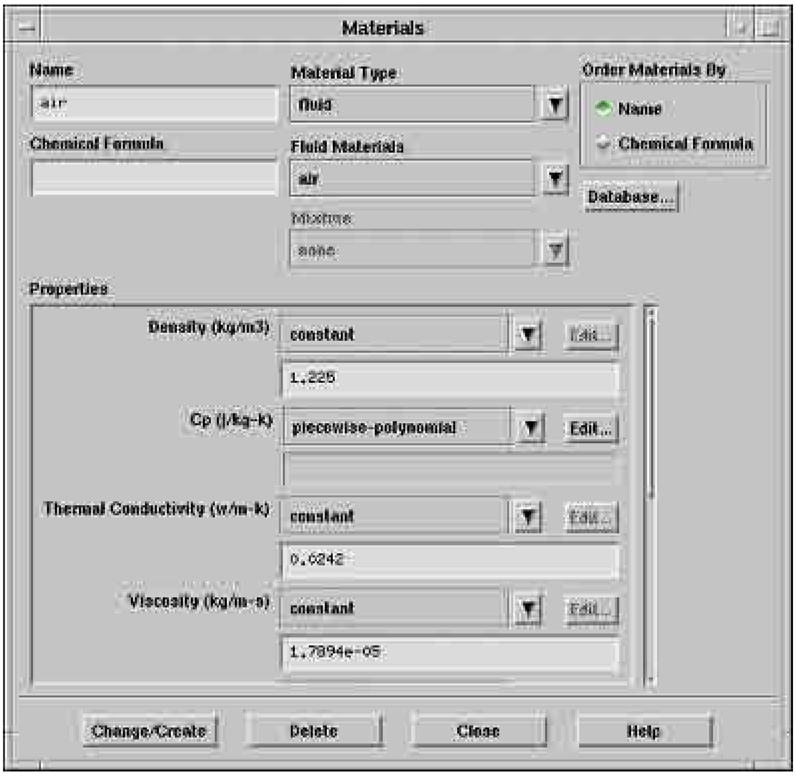


图 9-1 Materials（材料）面板

如果使用分离求解器，除非在模拟非定常流或者运动的固体区域，对于固体材料可以不需定义其密度和比热。对于定常流来说，固体材料列表中也会出现比热一项，但是该值只被用于焓的后处理程序中，计算时并不需要它。

9.1.1 材料类型

在 Fluent 软件中，流体和固体的物理性质与指定的材料（materials）相关——在计算时指定材料的名称便相应指定了该材料所对应的物性参数，这些物性参数随即作为边界条件被设定在相应的网格分区上。在计算组元输运问题时，需要定义一种包含所有组元在内的混合物材料。

除了混合物的物性参数需要设定外，还需要设定混合物中每种组元的物性参数，当然

这些混合物及其组元都属于流体材料。在使用弥散相模型时，还可以定义附加的物性参数。在 FLUENT 中有一些预先定义好的混合物材料，比如空气、煤油等等，如果在计算中需要改变这些材料中包含的组元，或改变组元的性质时，也可以通过 Materials（材料）面板进行修改。

物性参数值的设定可以从零开始，完全由用户设定，也可以通过编辑 FLUENT 软件中提供的材料数据库进行扩充（见 9.1.4 节）。需要特别强调的是，所有设定的物性参数会保存在 case 文件中，这样下次即使选用新的求解器计算，只要调用该 case 文件便不需要重新输入材料数据。

9.1.2 材料面板

材料面板如图 9-1 所示，使用这个面板可以创建新材料或复制、修改现有材料的物性参数。

单击 Define（定义）菜单下的 materials…（材料）便可以打开材料面板。

本节重点介绍一般物性参数的设定，对于随温度变化的物性参数的设定在下节中有详细讲解。

在默认情况下，materials（材料）列表仅包括一种流体物质 air（空气）和一种固体物质 aluminium（铝）。如果要计算的流体物质恰恰是空气，那么可以直接使用默认的物性参数，当然也可以修改后再使用。但绝大多数情况下，我们都需要从数据库中调用其他的物质或者定义自己的物质。举一个简单的例子，如果要研究的流体是水，那么可以有两种选择，一种是从数据库中调用现有的材料数据，并且根据需要决定是否修改其物性参数如密度、粘度等；另一种方法则是建立自己全新的水，并定义其所有的物性参数。

混合物物质只有在激活组元运输方程后才会出现。与此类似，惰性颗粒、液滴和燃烧颗粒在引入弥散相模型之前是不会出现的。但在一个混合物的数据从数据库中加载进来时，它所包含的所有组元的流体材料（组元）将会自动被复制。

1. 修改现有材料的材料属性

在绝大多数情况下都是从数据库中加载已有的材料数据，然后根据实际情况和计算需要修改材料的物性参数。修改物性参数的工作，必须在材料面板中完成，其步骤如下：

（1）在 Material Type（材料类型）下拉菜单中选择材料的类型（流体或固体）。

（2）根据（1）的选择，在 Fluid Materials（流体材料）或 Solid Materials（固体材料）的下拉菜单中选定要改变物性的材料。

（3）根据需要修改在 Properties（性质）框中所包含的各种物性参数。如果列出的物

性参数种类太多，则需要拖动滑动条以显示所有的物性参数。

在完成修改后，单击 Change/Create（修改/创建）按钮，新的物性参数便被设定了。

要改变其他材料的物性参数只要重复前面步骤便可，但每种材料的物性参数修改完毕后，都要单击 Change/Create（修改/创建）按钮进行确认。

2. 重命名已有材料

所有的材料都是通过材料名称和分子式来区分的。除自行创建的物质外，现有材料数据中只有材料名称可以改变，但却不能改变他们的分子式。

通过下面的步骤可以改变一种材料的名称：

（1）与改变物性参数一样，首先在 Material Type 下拉菜单中选择材料的类型（流体或固体）。

（2）根据第一步的选择，在 Fluid Materials 或 Solid Materials 的下拉菜单中选定要修改物性的材料。

（3）在面板顶部的 Name（名称）框中输入新的名称。

（4）单击 Change/Create（修改/创建）按钮，这时会弹出一个问题对话框，如图 9-2。



图 9-2 问题对话框

（5）选择 Yes（是）便完成了更改材料名称的工作。

采用同样的方法可以更改其他材料的名称，每次更改完成后单击 Change/Create（修改/创建）按钮进行确认。

3. 从数据库中复制材料

材料数据库中包含许多常用的流体、固体和混合物材料。调用这些材料的步骤非常简单，要做的仅仅是从数据库中把它们复制到当前的材料列表中。复制材料应采取以下步骤：

（1）单击材料面板的 Database...（数据库）按钮打开 Database Materials（材料数据库）面板，如图 9-3 所示。

（2）在 Material Type 下拉菜单中选择材料的类型（流体或固体）。

（3）根据（1）的选择，在 Fluid Materials 或 Solid Materials 的下拉菜单中选定要复制的材料，该材料的各种参数随即显示在属性（Properties）框中。

（4）可以拖动滚动条检查材料的所有参数。对于某些参数，除了用常值定义外，也可以用温度的函数形式加以定义。具体采用哪种定义形式，可以从物性右边的下拉列表中选择。具体设置方法，请参见后续章节的详细介绍。

（5）单击 Copy（复制）按钮，完成复制工作。

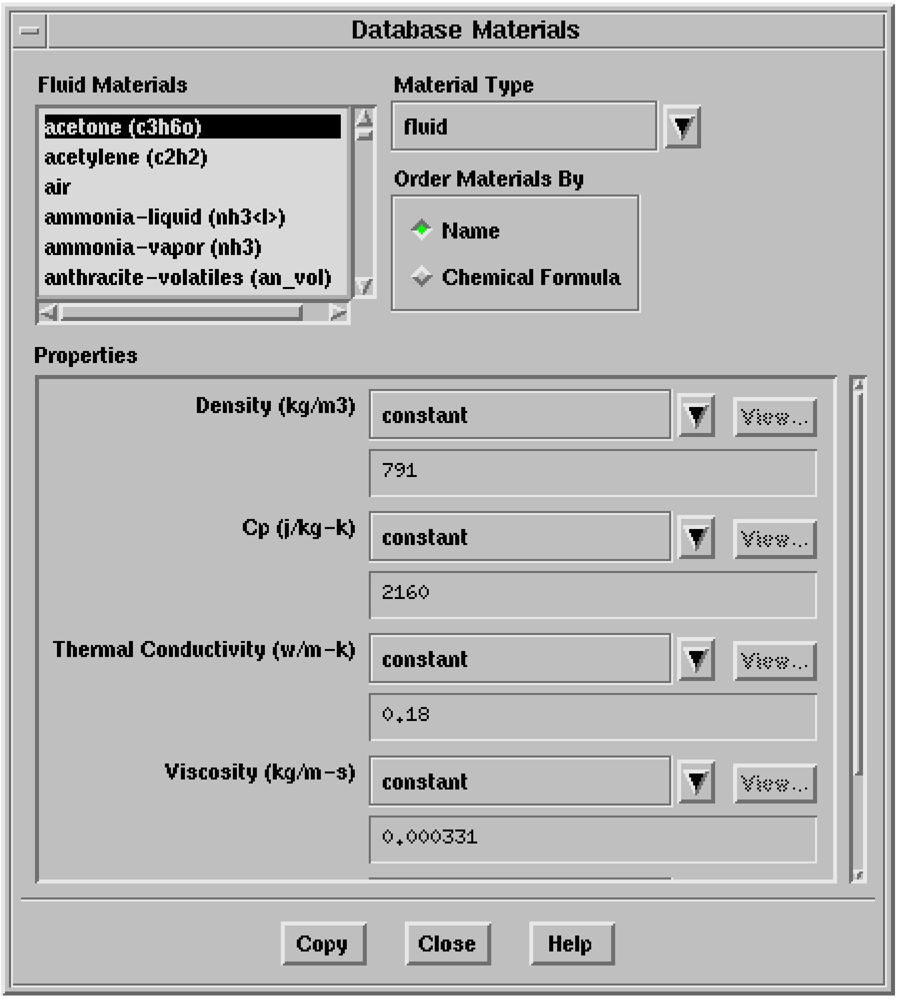


图 9-3 材料数据库面板

重复上述步骤可以复制其他的材料，复制工作全部完成时单击 close（关闭）按钮关闭材料数据库面板。

一旦将材料从数据库中复制完成，就可以如前面所讲的一样修改其物性参数，而且为了保持数据库的准确性，任何修改都不会影响原数据库里物性参数的数值。

4. 创立新的物质

在数据库中没有我们要用的材料的情况下，我们可以创建该物质。步骤如下：

（1）在 Material Type（材料类型）下拉列表中选择新材料的种类（流体或固体）。

（2）在 Name（名称）框中输入新材料的名称。

（3）在 properties（属性）框中，设定物质的属性.

（4）单击 Change/Creat（改变/创建）按钮。系统会弹出一个问题对话框，询问是否覆盖原有材料（即创立新材料前，材料面板显示的材料！）。选择 No（否），系统会添加新材料，并保持原材料。接着会弹出另外一个面板，要求输入新材料的分子式。如果已知则输入，否则保持空白并单击 OK 按钮。在上述步骤完成后面板会显示新的材料。

5. 保存材料及其物性参数

所有的材料及其物性参数都会保存在算例文件（.cas 文件）中。如果在新的任务中，读入该算例文件，则所有的材料及其物性参数都会被读入。

6. 删除材料

如果材料列表中有不再使用的材料，执行以下步骤可以将其删除：

（1）在材料类型下拉菜单中选择材料类型（流体或固体）。

（2）选择要删除的材料。

（3）单击 Delete（删除）按钮完成删除。

在当前列表中删除材料，不会影响到数据库数据。

7. 改变材料列表顺序

在默认状态下，当前列表及数据库中的材料都是按照名称顺序排列。如果想按照分子式排列，则在 Order Materials By（材料排序方式）菜单下，选择 Chemical Formula（分子式）即可。

需要说明的是，材料面板与材料数据库中材料的排列方式可以有所不同，比如材料面板中可以用名称排列，材料数据库中却可以用分子式排列。

9.1.3 用温度方程定义物性参数

材料属性可以用多项式、阶梯函数或者分段多项式等形式定义为温度的函数：

多项式函数：

$$\phi(T)=A_1+A_2T+A_3T^2+\ldots \tag{9-1}$$

阶梯函数：

$$\phi(T)=\phi_n+\frac{\phi_{n+1}-\phi_n}{T_{n+1}-T_n}(T-T_n) \tag{9-2}$$

其中$1\le n\le N$，N 为分段数。

分段多项式函数，对于时间段 1：

$$T_{\min,1}<T<T_{\max,1}:\phi(T)=A_1+A_2T+A_3T^2+\ldots \tag{9-3}$$

对于时间段 2：

$$T_{\min,2}<T<T_{\max,2}:\phi(T)=B_1+B_2T+B_3T^2+\ldots \tag{9-4}$$

注意：如果用温度的多项式函数或者分段多项式函数定义参数，其中的温度单位必须是 Kelvin（开氏温度）或者 Rankine（兰氏温度）。如果使用的是 Celsius（摄氏温度）或者 Kelvin（开氏温度）作为温度单位，则相应的多项式系数要用 Kelvin（开氏温度）进行设置。如果使用 Fahrenheit（华氏温度）或者 Rankine（兰氏温度）作为温度单位，则相应的多项式系数要用 Rankine（兰氏温度）单位进行设置。

除了上述三种函数形式外，一些材料性质还可以用其他函数形式定义，另外一些材料性质则只能用这三种函数中的一到二种函数加以定义。

1. 多项式函数的输入

执行以下步骤，便可以定义物性为温度的多项式函数：

（1）在材料面板物性参数右边的下拉菜单中选择 polynomial（多项式），系统会自动弹出 Polynomial Profile（多项式曲线）面板如图 9-4。

（2）确定多项式系数个数（Coefficients），最多可输入 8 个。默认的数字 1 定义方程为 0 阶的。物性将会是常数值，它等效于输入单个系数 A_1，输入 2 对应于 1 阶函数，物性将是温度的线性函数，依此类推。

（3）确定系数值，系数 1、2、3 与方程 9-1-1 中的 A_1、A_2、A_3 对应。图 9-4 中的输入

定义了如下方程：

$$\rho(T) = 1000 - 0.02T \tag{9-5}$$

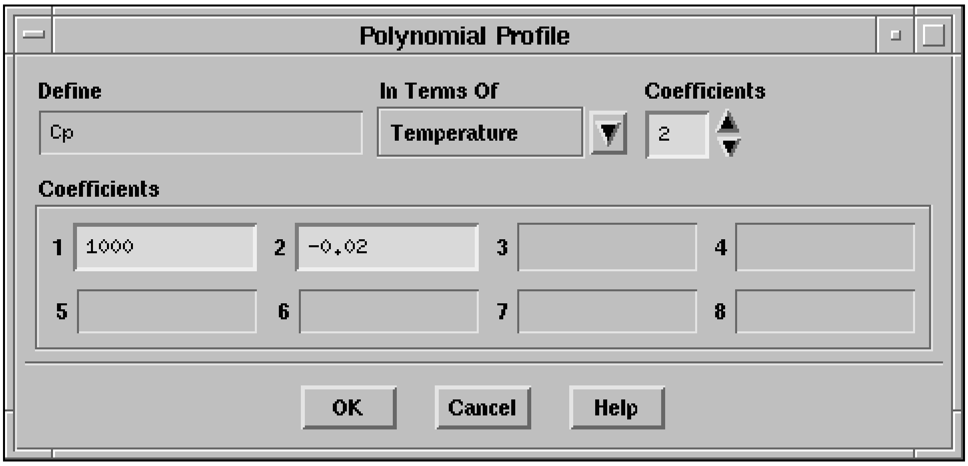


图 9-4 Polynomial Profile（多项式曲线）面板

2. 阶梯函数的输入

执行如下步骤，可以定义多段线性温度函数：

（1）材料面板中，在物性名称右边的下拉菜单中选择 piecewise-linear（阶梯函数），系统会自动弹出 Polynomial Profile（阶梯函数曲线）面板，如图 9-5。

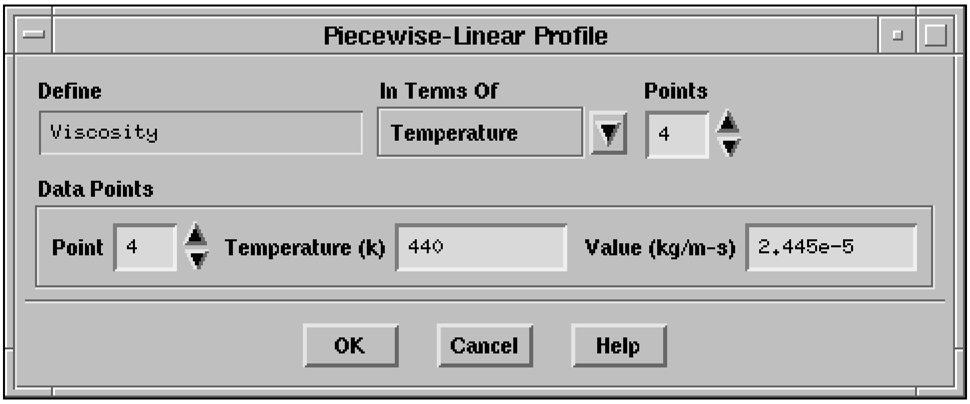


图 9-5 Piecewise-Linear Profile（阶梯函数曲线）面板

（2）定义阶梯函数分段的点数。

（3）在对应的数据点（data points）下，输入成对的数据值。首先输入点 1 的自变量和因变量数值，然后增加点的个数并且输入相应的数据值。数据值必须按温度递增的次序输入，Fluent 是不会自动排序的！每种属性最多可输入 30 个点。图 9-5 所示便是如图 9-6 的函数输入完成后的阶梯函数面板的样子。

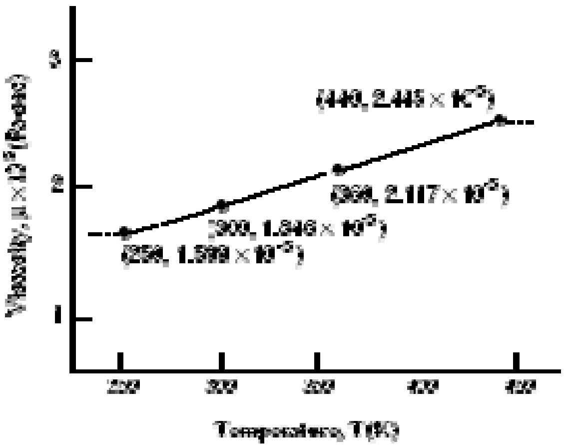


图 9-6 定义阶梯函数粘度 μ（T）

3. 分段多项式函数的输入

分段多项式函数的定义也非常简单，只要执行以下步骤便可以了。

（1）材料面板中，在物性名称右边的下拉菜单中选择 piecewise-polynomial（多段多项式），系统会自动弹出 Polynomial Profile（多段多项式曲线）面板如图 9-7。

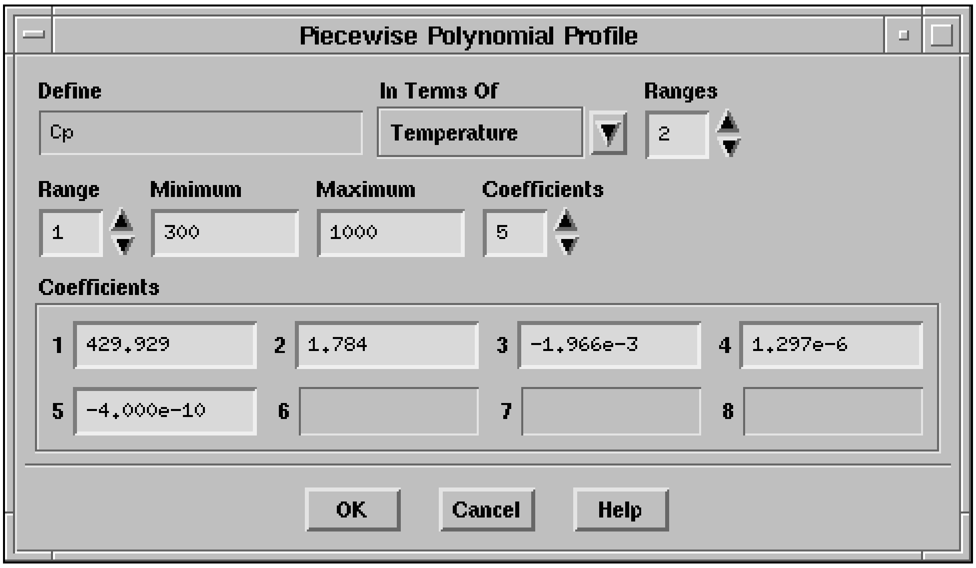


图 9-7 多段多项式函数输入面板

（2）指定分区（Ranges）的个数。对于方程（9-5），需要定义两个温度范围：

当$300 < T < 1000$时：

$$c_p(T) = 429.929 + 1.874T - 1.966\times10^{-3}T^2 + 1.297\times10^{-6}T^3 - 4.000\times10^{-10}T^4$$

当$1000 < T < 5000$时：

$$c_p(T) = 841.377 + 0.593T - 2.415\times10^{-4}T^2 + 4.523\times10^{-8}T^3 - 3.153\times10^{-12}T^4 \quad (9\text{-}6)$$

当然，也可以定义三个范围，但这些范围必须按温度升高的次序输入。

（3）在定义每个分区上的函数时，首先确定 Maximum（最大值）与 Minimum（最小值）温度，以及 Coefficients（系数）的个数（最多八个）。系数的个数决定了多项式的阶数。系统默认的 1 个系数，定义 0 阶函数，即函数为常值；系数个数等于 2 时，定义 1 阶线性方程，即函数将随温度呈线性变化，依此类推。

（4）输入系数值，即在 Coefficients（系数）栏下面的输入栏中分别输入多项式的系数值，其中的 1、2、3 等数字分别对应于多项式系数 A_1、A_2、A_3 等。

（5）将 Ranges（分区）序号增加为 2，并重复上述过程，定义分区 2 上的多项式。

（6）依此类推，定义其他分区上的多项式函数。

4. 检查与修改现有曲线

如果需要检查或改变已有多项式曲线的系数、数据点数值，或者温度范围，单击 Edit（编辑）按钮，Piecewise Polynomial Profile（分段多项式）面板便会打开，按照前面的输入方法便可以完成数据的检查与修改。

9.1.4 个性化材料数据库

材料数据库在计算机中的存储位置为：

path /Fluent.Inc/fluent6. **x**/cortex/lib/propdb.scm

式中，path 是 Fluent 的安装目录。变量 **x**，表示的是 Fluent 的版本，对于 FLUENT6.1 来说，x 值为 1。如果需要编辑材料数据库，使其包含经常使用的材料数据，可以按照如下步骤操作：

首先，从工作目录中复制文件 propdb.scm。用文本编辑软件（如写字板），按照下面的格式添加要新材料。在式中，可以复制线面的格式，仅仅改变需要变动的物性参数从而快速完成输入。输入格式如下：

```
（air    ←材料名称
 fluid    ←材料种类
 （chemical-formula . #f）    ←分子式
 （density  （constant . 1.225）   ←密度
 （premixed-combustion 1.225 300））  ←预混和燃烧
 （specific-heat  （constant . 1006.43））   ←热值
 （thermal-conductivity  （constant . 0.0242））   ←热传导系数
 （viscosity  （constant . 1.7894e-05）            ←粘度
 （sutherland 1.7894e-05 273.11 110.56）      ←索尔兰值
 （power-law 1.7894e-05 273.11 0.666））      ←幂函数
 （molecular-weight  （constant . 28.966））        ←分子量
 ）
（aluminum ←材料名称
 （solid）  ←材料种类
 （chemical-formula . al）  ←分子式
 （density  （constant . 2719））  ←密度
 （specific-heat  （constant . 871））  ←预混和燃烧
 （thermal-conductivity  （constant . 202.4））  ←热传导系数
```

（formation-entropy （constant . 164448.08）） 熵值

）

编辑完成后，在 Fluent 工作目录中加载修改过的 propdb.scm 文件，而不是原有的数据库。这样，个性化的数据库便出现在材料列表中了。

如果希望在其他计算中也使用修改过的数据库文件，只要把修改过的 propdb.scm 文件复制到 cortex/lib 目录便可以了。建议复制之前备份原有数据库。

9.2 密度

Fluent 也为密度提供了多种定义方式，可以是常数值，也可以是温度的函数，或者是组元的函数。所有的定义方式也都需要在 Materials（材料）面板中完成。

9.2.1 为不同的流动区域定义密度

在不同的流动区域上，可以定义不同的密度变化规律。在定义密度的过程时应该遵循下面几条简单原则：

（1）对于可压缩流体，应该采用理想气体定律。

（2）对于不可压缩流体，有四种定义密度的方案：

1）如果不希望将密度定义为温度的函数，可以直接将其定义为常数值。

2）如果压力的变化极其微小，但希望用理想气体关系定义密度与温度的关系时，可以选择不可压理想气体定律。

3）如果密度函数只有温度一个自变量，则可以用多项式函数、多段多项式函数或阶梯函数等形式定义二者之间的关系。

4）在计算自然对流等温度变化很小的问题时可以采用 Boussinesq 模型。

（3）多重区域混合密度关系。

在流场中含有多块流动区域，而且各流动区域使用不同的材料时，应该注意下列问题：

1）在采用分离求解器时，如果有一种物质使用了理想气体定律，则所有的物质都必须使用此定律。耦合求解器没有这个限制。

2）在计算模型中，操作压强和操作温度是唯一的。就是说，如果有二种以上的材料使用了理想气体定律，则它们的操作压强都是一样的；如果有二种以上的材料使用了 Boussinesq 模型，则其操作温度都是一样的。

9.2.2 采用常值密度

如果流体的密度值为常数，首先确定 Materials（材料）面板中 Density（密度）右边的下拉列表中，Constant（常数）为当前选项，然后在输入栏中输入数值就可以了。

9.2.3 采用 Boussinesq 近似模型

在 Material（材料）面板中，从 Density（密度）右边的下拉列表中选择 boussinesq 选项，指定密度定义方式为 boussinesq 近似。为了正确计算密度，要正确定义 Thermal Expansion Coefficient（热膨胀系数）和相应的操作温度。采用 boussinesq 模型的原因是：与将密度定义为温度的函数相比，boussinesq 模型的收敛速度更快，因为该模型在计算过程中把密度作为常数，而只对动量方程中的浮力项应用下面的关系式：

$$(\rho-\rho_0)g \approx -\rho_0\beta(T-T_0)g \quad (9\text{-}7)$$

式中 ρ_0 是常数值， T_0 是操作温度， β 是热膨胀系数。

9.2.4 采用密度为温度函数的定义方式

当研究的问题中涉及到热传导时，可以把密度定义为温度的函数，定义形式包括阶梯函数、分段多项式函数和多项式函数，具体设置方法请参见 9.1.3 节中的内容。

9.2.5 不可压缩理想气体定律

在 Fluent 软件中，如果用理想气体定律定义不可压流动的密度，求解器将会按照下式计算密度

$$\rho=\frac{p_{op}}{\frac{R}{M_w}T} \quad (9\text{-}8)$$

式中 R 是普适气体常数，M_W 是气体分子量，P_{op} 是操作压强。在该式中，密度只与操作压强有关，而与当地压强场无关。理想不可压气体密度的输入步骤如下：

（1）首先在 Material（材料）面板中 Density（密度）右侧的下拉列表中选择 incompressible-ideal-gas（不可压理想气体）。需要注意的是，对每一种采用理想气体假设的材料都必须单独进行这项设置。

（2）在 Operating Conditions（操作条件）面板中设定操作压强。执行下列菜单操作打开 Operating Conditions（操作条件）面板：

Define -> Operating Conditions...

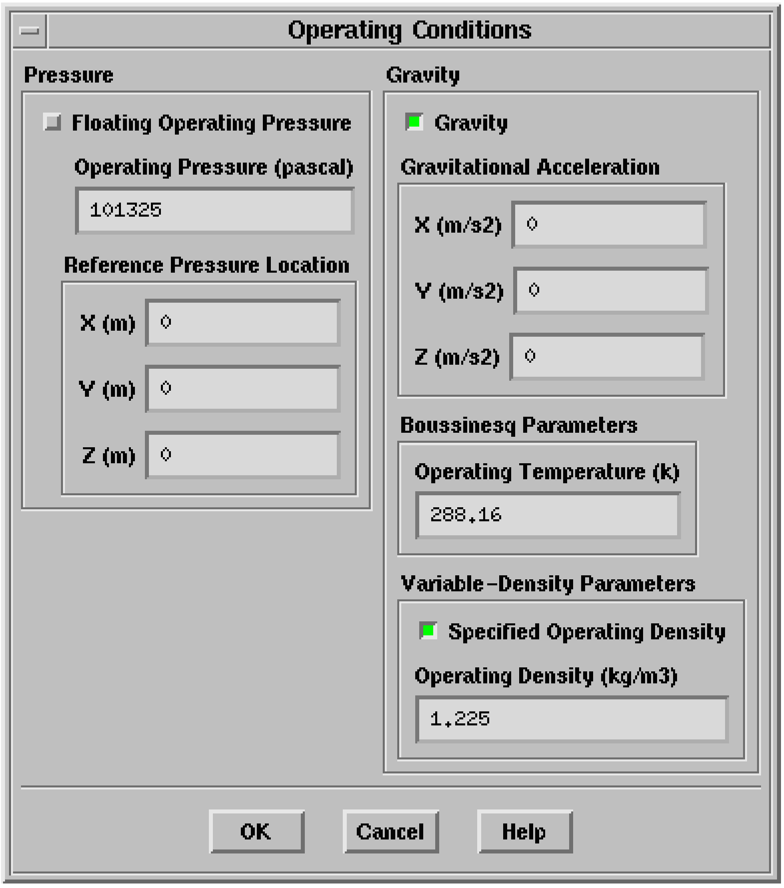


图 9-8 Operating Conditions（操作条件）面板

（3）操作压强的输入对于应用理想气体定律计算密度至关重要。在缺省设置中，操作

压强的数值为一个标准大气压，即 101325 Pa。

（4）在 Material（材料）面板中设定分子量。如果计算中包含组元输运，则为每个组元设置分子量。

9.2.6 可压缩气体的理想气体定律

对于可压缩气体，气体定律有如下形式：

$$\rho = \frac{p_{op} + p}{\frac{R}{M_w}T} \tag{9-9}$$

式中 p 为当地表压，p_{op} 为操作压强。

可压缩理想气体密度的输入步骤如下：

（1）在 Materials（材料）面板 Density（密度）右侧的下拉列表中选择 ideal-gas（理想气体）。注意每种物质密度的输入都必须单独完成。

（2）在 Operating Conditions（操作条件）面板中设定操作压强。

（3）在 Materials（材料）面板中设定分子量或者每个组元的分子量。

9.2.7 多组元混合物的组合密度

如果求解组元输运方程，则需要为混合物和组元设定相关属性。要定义与混合物组元相关的密度，可按下列步骤操作：

（1）选择密度定义方法。

密度定义方法应该在材料面板 Density（密度）右边下拉列表中选择。与非理想气体混合物对应的方法为混合物的 volume-weighted-mixing-law（体积加权混合定律）方法。与可压流动对应的方法是混合物的 Idea-gas（理想气体）方法。如果使用理想气体定律模拟不可压流动，则选择混合材料的不可压理想气体选项。

（2）点击 change/creat（改变/创建）按钮。

（3）如果选择 volume-weighted-mixing-law（体积加权混合律）定义组成混合物的每一种组元的密度，可以将每一个组元密度定义为常数或者温度的函数。

（4）如果选择 user-defined-mixing-law（用户定义混合律），则可以用 UDF（用户自定义函数）形式定义密度，具体方法请参见本书后续章节中与 UDF 相关的内容。

若计算非理想气体混合物，FLUENT 以下面公式计算混合物的密度：

$$\rho = \frac{1}{\sum_i \frac{Y_i}{\rho_i}} \qquad (9\text{-}10)$$

式中Y_i是组元 i 的质量浓度，ρ_i是组元 i 的密度。

对于可压流理想气体定律的形式为：

$$\rho = \frac{p_{op} + p}{RT\sum_i \frac{Y_i}{M_i}} \qquad (9\text{-}11)$$

式中 p 是当地表压，R 是普适气体常数，Y_i是组元 i 的质量浓度，M_i是组元 i 的分子量，p_{op}是操作压强。

若所计算的是不可压流动的理想气体定律的密度，FLUENT 用下面公式计算混合气体的密度：

$$\rho = \frac{p_{op}}{RT\sum_i \frac{Y_i}{M_i}} \qquad (9\text{-}12)$$

式中R是普适气体常数，Y_i是组元 i 的质量浓度，M_i是组元 i 的分子量，p_{op}是操作压强。在这种形式中，密度只与操作压强有关而与当地表压无关。

9.3 粘度

与密度定义相似，FLUENT 中也有几种定义流体粘度的方式：

（1）定义常数粘度。

（2）定义温度和（或）组元相关粘度。

（3）用动力学理论定义粘度。

（4）定义非牛顿流体粘度。

（5）用 UDF（用户自定义函数）定义粘度。

本节将讲解上述每一种粘度定义方法的输入过程及相关的物理模型。在所有计算中，粘度都是在 Materials（材料）面板中定义的。

FLUENT 中动力粘度（μ）的单位是 kg/m-s（国际单位制）或 lb-m/ft-s（英制单位）。FLUENT 不需要输入运动粘度（ν）。

9.3.1 输入常数粘度

如果将流体粘度定义为常数，首先在 Materials（材料）面板中 Viscosity（粘度）右边的下拉列表中选择 constant（常数），然后输入流体的粘度值即可。系统默认的流体为空气，其粘度默认值为 1.7894×10^{-5} kg/m-s。

9.3.2 输入温度函数粘度

如果所计算的问题包括热传导，可以将粘度定义为温度的函数。FLUENT 共提供了五种函数形式用于温度定义，在计算中可以根据需要从 Viscosity（粘度）旁边的下拉列表中选择使用。这五种函数形式包括阶梯函数、分段多项式、多项式、Sutherland 公式和幂律形式。前三种形式的设置方法请参见 9.1.3 节中的内容，这里只介绍后面二种方法：

（1）Sutherland 公式。

Sutherland （1893）使用理想化的分子间作用力势函数推出 Sutherland 粘度公式，公式由二或三个系数确定。

由二个系数确定的 Sutherland 公式为：

$$\mu=\frac{C_1T^{3/2}}{T+C_2} \tag{9-13}$$

式中粘度的单位为 kg/m-s，温度的单位为 K，C_1和C_2是系数。对于常温常压下的空气来说，C_1 = 1.458×10^{-6} kg/m-s-$K^{1/2}$，C_2 = 110.4 K.

由三个系数确定的 Sutherland 公式为：

$$\mu = \mu_0 \left(\frac{T}{T_0} \right)^{3/2} \frac{T_0 + S}{T + S} \qquad (9\text{-}14)$$

式中 μ 为粘度，单位为 kg/m-s；T 是静温，单位为 K；μ_0 是参考粘度，单位为 kg/m-s；T_0 是参考温度，单位为 K；S 是反映气体特征的有效温度，被称为 Sutherland 常数，单位为 K。在中等温度和压强下，$\mu_0 = 1.7894\times10^{-5}$ kg/m-s，$T_0 = 273.11$ K，S = 110.56 K。

在粘度右边的下拉列表中选择 sutherland 后，Sutherland Law（Sutherland 定律）面板就会打开，然后选择二系数或者三系数方法。需要注意的是，二系数方法必须使用国际标准单位。对于二系数方法，设定 C_1 和 C_2 即可。对于三系数方法，需要设定参考粘度 μ_0、参考温度 T_0 以及有效温度 S。

（2）幂律。

稀薄气体粘度的另一个近似模型是幂律公式。对于中等温度的稀薄气体，幂律形式在精度上比 Sutherland 公式略逊一筹。

二系数的幂律粘度定律形式为：

$$\mu = BT^n \qquad (9\text{-}15)$$

式中 μ 是粘度，单位为 kg/m-s；T 是静温，单位为 K；B 是无量纲系数。对于中等温度和压力的空气，$B = 4.093\times10^{-7}$，n = 2/3。

三系数的幂律粘度定律的形式为：

$$\mu = \mu_0 \left(\frac{T}{T_0} \right)^n \qquad (9\text{-}16)$$

式中 μ 是粘度，μ_0 是参考粘度，单位均为 kg/m-s；*T* 是静温，T_0 为参考温度，单位均为 K。对于适当压力和温度的空气，$\mu_0 = 1.716\times10^{-5}$ kg/m-s，$T_0 = 273$ K，$n = 2/3$。

在 Viscosity（粘度）右边的下拉列表中选择 power-law（幂律），Power Law（幂律）面板随即打开，然后在面板上选择 Two Coefficient Method（二系数）或 Three Coefficient Method（三系数）方法。需要注意的是，二系数方法必须使用国际标准单位。对于二系数方法，设定 *B* 和温度指数 *n*。对于三系数方法，设定参考粘度 μ_0，参考温度 T_0 以及温度指数 *n*。

9.3.3 用动力学理论定义粘度

如果使用气体定律，可以选择使用动力学理论定义流体粘度：

$$\mu = 2.67 \times 10^{-6} \frac{\sqrt{MT}}{\sigma^2 \Omega_\mu} \tag{9-17}$$

式中 μ 粘度的单位是 kg/m-s，T 的单位为 K，M 为分子量，σ 的单位是埃，$\Omega_\mu = \Omega_\mu(T^*)$，式中：

$$T^* = \frac{T}{(\varepsilon/k_B)} \tag{9-18}$$

在 Materials（材料）面板中 Viscosity（粘度）右边的下拉列表中选择 kinetic-theory（动力学理论），则面板将自动展开出现新的输入栏。Lennard-Jones 参数 σ 和 ε / k_B 在 FLUENT 中分别被称为 L-J Characteristic Length（L-J 特征长度）和 L-J Energy Parameter（L-J 能量参数）。在输入这两个 Lennard-Jones 参数后，就可以用方程（9-3-8）唯一确定粘度与温度之间的关系，也就是说在计算中只要已知温度，就可以求出粘度。

9.3.4 多组元混合物的组元相关粘度

在计算多组元混合物的流动时，可以将混合物的粘度定义为各组元粘度的函数，也可以将混合物粘度定义为常数或定义为温度的函数。

在 FLUENT 中定义组元相关粘度的步骤如下：

（1）在 Viscosity（粘度）右边的下拉列表中选择 mass-weighted-mixing-law（质量加权混合律）。如果在密度定义中使用了理想气体定律，则选择 ideal-gas-mixing-law（理想气体混合定律）。如果准备用自定义函数定义粘度，则选择 user-defined（用户自定义），或选择 user-defined-mixing-law（用户自定义混合律）。

（2）点击 Change/Create（修改/创建）按钮。

（3）定义混合物中每个组元的粘度。定义方法可参考 9.3.1 到 9.3.3 几个小节的内容。

（4）user-defined-mixing-law（用户自定义混合律）是采用 UDF 来定义混合物的粘度，其基本定义方法也是先分别定义混合物中的各种组元，再用各组元的物性参数计算出混合物的粘度。user-defined（用户自定义）则是以混合物粘度参数为对象进行定义的。相关内容请参见 UDF 相关章节。

在使用理想气体定律时，混合物的粘度是用分子运动的动力学理论计算出来的：

$$\mu = \sum_i \frac{X_i \mu_i}{\sum_j X_i \phi_{ij}} \tag{9-19}$$

式中：

$$\phi_{ij} = \frac{\left[1 + \left(\frac{\mu_i}{\mu_j}\right)^{\frac{1}{2}} \left(\frac{M_j}{M_i}\right)^{\frac{1}{4}}\right]^2}{\left[8\left(1 + \frac{M_i}{M_j}\right)\right]^{\frac{1}{2}}} \tag{9-20}$$

式中 X_i 是组元 i 的摩尔浓度。

非理想气体混合物粘度则简单地取为各组元粘度的质量平均值：

$$\mu = \sum_i Y_i \mu_i \tag{9-21}$$

Y_i 是组元 i 的质量浓度，μ_i 是组元 i 的粘度。

9.3.5 非牛顿流体的粘度

对于不可压牛顿流体来说，剪切应力与变形速率张量成正比：

$$\tau = \mu \dot{S} \tag{9-22}$$

式中：

$$\dot{S} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \tag{9-23}$$

粘度 μ 与 $\dot{S}$ 无关。

对于非牛顿流体来说，在引入一个非牛顿流体粘度 η 后，剪切应力 τ 可以仿照牛顿流体的本构关系（9-3-12）写成变形张量 $\dot{S}$ 的函数：

$$\tau = \eta(\dot{S})\dot{S} \tag{9-24}$$

在一般情况下，η 应该是变形张量 $\dot{S}$ 的三个不变量的函数。在 FLUENT 中则将其简化为仅仅是剪切速率 $\dot{\gamma}$ 的函数，剪切速率 $\dot{\gamma}$ 的定义为：

$$\dot{\gamma} = \sqrt{\dot{S} : \dot{S}} \qquad (9\text{-}25)$$

FLUENT 提供了四种模型用于非牛顿流的计算，即幂律模型、Carreau 模型、Cross 模型和 Herschel-Bulkley 模型。下面分别介绍这四种模型：

（1）幂律模型。

非牛顿流体幂律粘度的形式为：

$$\eta = k\dot{\gamma}^{n-1} e^{T_0 / T} \qquad (9\text{-}26)$$

FLUENT 允许为幂律函数设置上下限：

$$\eta_{\min} < \eta = k\dot{\gamma}^{n-1} e^{T_0 / T} < \eta_{\max} \qquad (9\text{-}27)$$

式中 k、n、T_0、$\eta_{\min}$ 和 $\eta_{\max}$ 为输入参数。k 是流体平均粘度（稠度指数）的量度，n 是与满足幂律的牛顿流体背离程度的量度，T_0 是参考温度，$\eta_{\min}$ 和 $\eta_{\max}$ 分别为幂律的上下限。当从幂律模型计算出的粘度值超出上下限范围时，Fluent 就用 $\eta_{\min}$ 和 $\eta_{\max}$ 分别取代超出上限和下限的值。

n 的值决定了流体的类型：$n = 1$ 为牛顿流体；$n > 1$ 为剪切增厚流体（胀流型流体）；$n<1$ 为剪切变薄流体（仿塑胶流体）。

在 Viscosity（粘度）右边的下拉列表中选择 non-newtonian-power-law（非牛顿幂律），则 Non-Newtonian Power Law（非牛顿幂律）面板随即打开。输入项包括 Consistency Index k（稠度指数 k）、Power-Law Index n（幂律指数 n）、Reference Temperature T_0（参考温度 T_0）、Mininum Viscosity Limit $\eta_{\min}$（最小粘度 $\eta_{\min}$）和 Maximum Viscosity Limit $\eta_{\max}$（最大粘度 $\eta_{\max}$）。对于温度无关的粘度值，应该将 T_0 设置为 0。如果计算中不包含能量方程，FLUENT 用温度的缺省值 273K 进行幂律粘度计算。

（2）用于仿塑胶计算的 Carreau 模型。

非牛顿流体粘度的幂律模型给出的粘度 η 随剪切速率 $\dot{\gamma}$ 的变化关系为：$\dot{\gamma}$ 趋近于 0 时，

η趋近于η_0；$\dot{\gamma}$趋近于无穷大时，η趋近于η_∞[1]。

Carreau 模型则使用曲线拟合将牛顿流体和剪切变薄（n<1）非牛顿流体结合在一起，从而达到模拟更大范围流体粘度的目的。Carreau 模型的粘度定义为：

$$\eta = \eta_\infty + (\eta_0 - \eta_\infty)\left[1 + \left(\dot{\gamma}\lambda e^{T_0/T}\right)^2\right]^{(n-1)/2} \qquad (9\text{-}28)$$

式中参数 n、λ、T_0、η_0和η_∞对不同流体的取值有所不同。λ是时间常数，n是幂律指数（详见非牛顿粘度的幂律模型），T_0为参考温度，η_0和η_∞分别为零剪切和无穷剪切粘度。

在 Viscosity（粘度）右边的下拉列表中选择 carreau，Carreau Model（Carreau 模型）面板随即打开。此时可以输入时间常数λ、幂律指数 n、参考温度 T_0、零剪切粘度η_0和无穷剪切粘度η_∞。

（3）Cross 模型。

Cross 模型粘度定义为：

$$\eta = \frac{\eta_0}{1 + (\lambda \dot{\gamma})^{1-n}} \qquad (9\text{-}29)$$

式中η_0为零剪切速率粘度，λ为时间系数，n为幂律指数。Cross 模型通常用于计算低剪切速率粘度。

在 Viscosity（粘度）右边的下拉列表中选择 cross，就可以打开 Cross Model（Cross 模型）面板。可以输入的参数包括时间常数λ、幂律指数 n、参考温度 T_0和零剪切粘度η_0。

（4）用于计算 Bingham 塑胶粘度的 Herschel-Bulkley 模型。

Herschel-Bulkley 模型用于模拟在剪切应变为零时，剪切应力不为零的流体的粘度，这种模型对粘度的定义为：

$$\eta = \frac{\tau_0 + k[\dot{\gamma}^n - (\tau_0/\mu_0)^n]}{\dot{\gamma}} \qquad (9\text{-}30)$$

---

[1] η_0和η_∞分别为粘度的下限和上限。

式中 k 为稠度因子，n 为幂律指数。

图 9-9 为 Herschel-Bulkley 模型中剪切应力 τ 与剪切速率 $\dot{\gamma}$ 的关系。

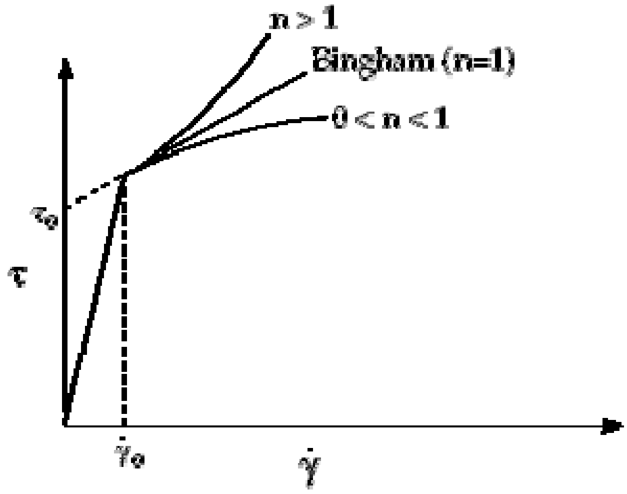


图 9-9 剪切应力与剪切速率关系曲线

在 Viscosity（粘度）右边的下拉列表中选择 herschel-bulkley，则 Herschel-Bulkley 模型面板将自动打开。面板中可以输入的参数包括 Consistency Index k（稠度指数 k）、Power-Law Index n（幂律指数 n）、Yield Stress Threshold τ_0（屈服应力阈值 τ_0）、Yielding Viscosity μ_0（屈服粘度 μ_0）。

9.4 导热系数

当计算中涉及热传导时，必须定义导热系数。在求解能量方程和粘性流动时也需要定义导热系数。

FLUENT 提供了几种定义导热系数的方法，即常数导热系数、温度和（或）组元相关导热系数、用分子动力论定义导热系数、用自定义函数定义导热系数和各向异性导热系数[2]。

导热系数的定义也是在 Materials（材料）面板中定义。

[2] 仅适用于固体材料。

导热系数所使用的单位包括 W/m-K（国际单位）和 BTU/hr-ft-^{o}R（英制单位）。

9.4.1 常数导热系数的输入

如果要定义常数导热系数，首先在 Materials（材料）面板 Thermal Conductivity（导热系数）右边的列表中选择 constant（常数），然后输入导热系数的值即可。

对于缺省设置的流体——空气，其导热系数默认为 0.0242 W/m-K。

9.4.2 作为温度函数的导热系数

可以将导热系数定义为温度的函数，FLUENT 提供了三种类型的函数，即阶梯函数、分段多项式和多项式，详细设置方法请参见 9.1.3 中的内容。

9.4.3 使用分子动力学理论定义导热系数

如果所计算的流体为气体，则可以选择用分子动力学理论定义导热系数：

$$k=\frac{15}{4}\frac{R}{M}\mu\left[\frac{4}{15}\frac{c_pM}{R}+\frac{1}{3}\right] \qquad (9\text{-}31)$$

式中 R 是普适气体常数，M 是分子量，μ 为动力粘度，c_p 为定压比热。

在 Materials（材料）面板中的 Thermal Conductivity（导热系数）右边的下拉列表中选择 kinetic-theory（动力学理论），求解器就会使用上面的方程来计算导热系数。

9.4.4 多组元混合物的组元相关导热系数

如果流场中包含多种化学组元（多组元流动），则可以选择定义组元相关的导热系数。

定义多组元导热系数的步骤如下：

（1）在计算多组元混合物时，可以在 Thermal Conductivity（导热系数）右边的下拉列表中选择 mass-weighted-mixing-law（质量加权混合律）定义导热系数。如果混合物为理想气体，则应该选择 ideal-gas-mixing-law（理想气体混合律）。

如果用理想气体混合律计算混合物的导热系数，则必须同时使用 ideal-gas-mixing-law

（理想气体混合律）或者 mass-weighted-mixing-law（质量加权混合律）计算粘性，因为只有这两种方法可以用于指定多组元混合物的粘性。

（2）点击 Change/Create（改变/创建）按钮。

（3）定义混合物中每个组元材料的导热系数，定义方式包括常数、温度相关函数和分子动力论等等。

（4）若使用理想气体定律，解算器会在分子运动论的基础上计算混合物的导热系数：

$$k = \sum_i \frac{X_i k_i}{\sum_j X_j \phi_{ij}} \tag{9-32}$$

式中：

$$\phi_{ij} = \frac{\left[1 + \left(\frac{\mu_i}{\mu_j}\right)^{\frac{1}{2}} \left(\frac{M_j}{M_i}\right)^{\frac{1}{4}}\right]^2}{\left[8\left(1 + \frac{M_i}{M_j}\right)\right]^{\frac{1}{2}}} \tag{9-33}$$

式中 X_i 是组元 i 的摩尔浓度。.

对于非理想气体，混合物的导热系数与各组元的导热系数之间的关系为：

$$k = \sum_i Y_i k_i \tag{9-34}$$

式中 Y_i 为组元 i 的质量浓度，k_i 为组元 i 的导热系数。

多组元混合物的导热系数也可以用 UDF（用户自定义函数）来定义，其选项为 user-defined（用户定义）或 user-defined-mixing-law（用户定义混合律），详细设置方法请参见 UDF 的相关章节。

9.4.5 固体的各向异性导热系数

FLUENT 中各向异性导热系数选项解决固体的热传导方程时，所使用的导热系数被指定为一个矩阵。热流量为：

$$q_i = -k_{ij} \frac{\partial T}{\partial x_j} \tag{9-35}$$

式中 k_{ij} 为导热系数矩阵，各向异性材料导热系数的定义方式有五种，即双轴导热系数、

正交各向异性、圆柱各向异性、用户自定义各向异性和一般各向异性导热系数，下面将分别予以介绍。各向异性导热系数只能在分离求解器中使用，而不能用在耦合求解器中。

（1）双轴导热系数。

双轴导热系数主要用于壳体导热模型计算，在 Thermal Conductivity（导热系数）右边的下拉列表中选择 biaxial（双轴），就可以使用这种模型进行计算。

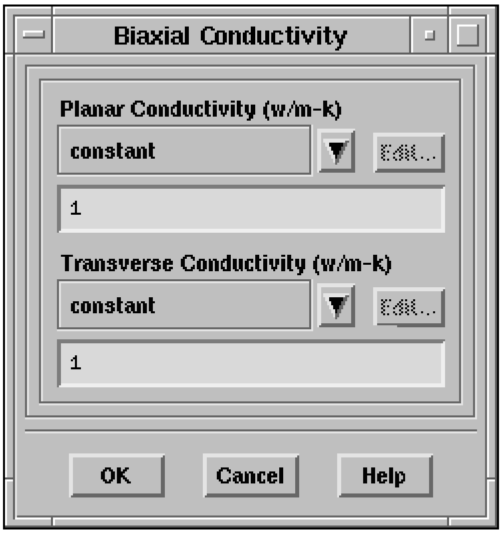


图 9-10　　Biaxial Conductivity（双轴导热系数）面板

这个模型中可以定义的两个方向的导热系数，一个是垂直于壳体表面方向的导热率，Transverse Conductivity（横向导热率）；另一个是沿壳体表面方向的导热率 Planar Conductivity（平面导热率）。两个参数均可以用常数、多项式、阶梯函数和分段多项式等形式来定义。

（2）正交各向异性导热系数

采用正交各向异性导热系数定义，首先需要指定三个主轴方向$\left(\hat{e}_\xi, \hat{e}_\eta, \hat{e}_\zeta\right)$上的导热系数$\left(k_\xi, k_\eta, k_\zeta\right)$，然后可以用下式唯一确定热传导矩阵：

$$k_{ij} = k_\xi e_{\xi i} e_{\xi j} + k_\eta e_{\eta i} e_{\eta j} + k_\zeta e_{\zeta i} e_{\zeta j} \quad (9\text{-}36)$$

在 Materials（材料）面板中 Thermal Conductivity（导热系数）右边的下拉列表中选择 orthotropic（正交各向异性），Orthotropic Conductivity（正交各向异性导热系数）面板随即打开，如图 9-4-1。

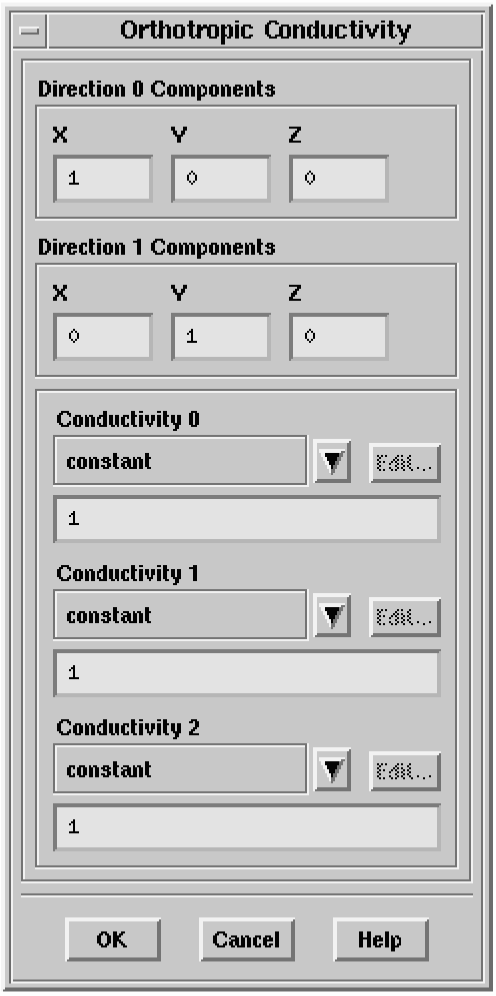


图 9-11 Orthotropic Conductivity（正交各向异性导热系数）面板

因为主轴方向$(\hat{e}_\xi, \hat{e}_\eta, \hat{e}_\zeta)$是相互正交的，所以对于三维问题只需指定前两个方向，就可以完全确定主轴方向。$\hat{e}_\xi$方向对应于 Orthotropic Conductivity（正交各向异性）面板上的 0 方向，在 Direction 0 Components（0 方向分量）中用$\hat{e}_\xi$在直角坐标系中的三个分量定义。$\hat{e}_\eta$对应于 1 方向，在 Direction 1 Components（1 方向分量）中用$\hat{e}_\eta$在直角坐标系中的三个

分量定义。

三个主轴方向的导热系数 $\left(k_{\xi}, k_{\eta}, k_{\zeta}\right)$ 则分别在 Conductivity 0（0 方向导热系数）、Conductivity 1（1 方向导热系数）和 Conductivity 2（2 方向导热系数）中定义。

在二维问题中只要给定 $\left(k_{\xi}, k_{\eta}\right)$ 和方向 $\hat{e}_{\xi}$ 即可。

（2）圆柱各向异性导热系数。

圆柱各向异性导热系数定义适用于材料导热特性在轴向、径向和周向有明显不同的材料。在 Materials（材料）面板中 Thermal Conductivity（导热系数）旁边的下拉列表中选择 cyl-orthotropic（圆柱正交各向异性），则 Cylindrical Orthotropic Conductivity（圆柱正交各向异性）面板随即打开。

在这个面板中需要定义圆柱坐标系的原点（Axis Origin）和轴向方向（Axis Direction），定义方法仍然是用直角坐标系中的坐标和方向余弦来进行定义。在圆柱坐标系的原点和方向被确定后，再分别输入径向导热系数（Radial Conductivity）、周向导热系数（Tangential Conductivity）和轴向导热系数（Axial Conductivity）的值，就可以唯一确定导热系数矩阵。

在二维计算中，仅需定义坐标系原点、径向导热系数和周向导热系数，就可以唯一确定导热系数矩阵，而坐标系的轴向方向被定义为垂直于屏幕的方向。

（3）用户自定义各向异性导热系数。

各向异性材料的导热系数矩阵也可以通过用户自定义函数（UDF）来定义，其中必须用到 DEFINE_PROPERTY 宏命令，详细情况请参见 UDF 相关章节。

（4）一般各向异性导热系数

热传导矩阵的一般式可以写作：

$$k_{ij} = k\hat{e}_{ij} \tag{9-4-9}$$

其中 k 是导热系数，$\hat{e}_{ij}$ 是矩阵（对于二维问题是 2×2 的矩阵，对于三维问题是 3×3 的矩阵）。

在 Materials（材料）面板中的 Thermal Conductivity（导热系数）右边的下拉列表中选择 anisotropic（各向异性），可以打开 Anisotropic Conductivity（各向异性导热系数）面板，如图 9-4-2。

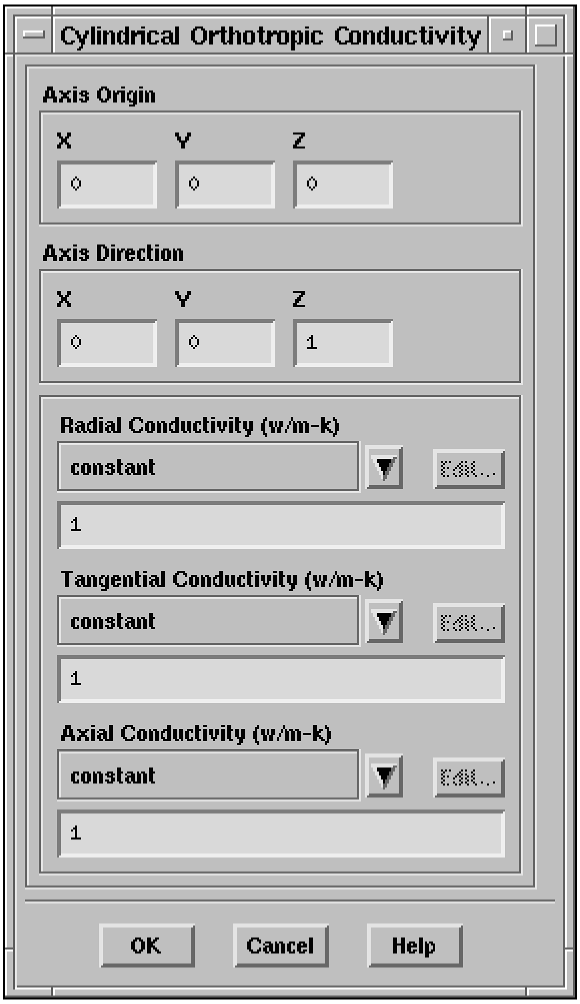


图 9-11 Cylindrical Orthotropic Conductivity（圆柱正交各向异性导热系数）面板

$\hat{e}_{ij}$ 可以是非对称矩阵，可以在 Matrix Components（矩阵分量）中指定矩阵的各分量。

k 可以用常数、多项式、阶梯函数或分段多项式等形式进行定义，也可以用自定义函数进定义。

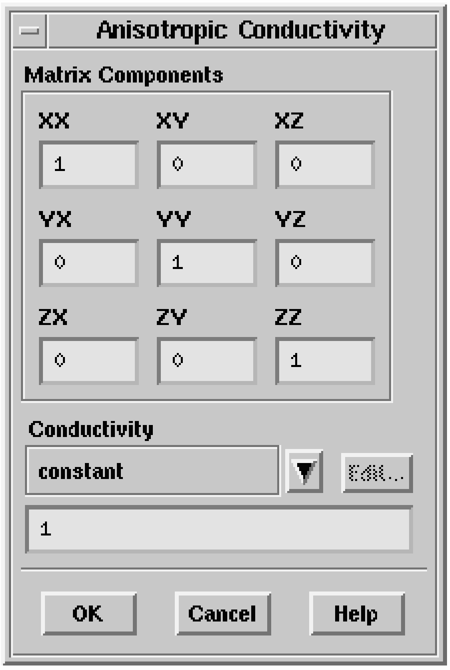


图 9-12　Anisotropic Conductivity（各向异性导热系数）面板

9.5 比热

在计算能量方程时必须定义比热。FLUENT 中所说的比热，指的是定压比热 c_p，单位为 J/kg-K（国际单位制）或者 BTU/lbm-R（英制）。

FLUENT 提供了三种定义比热的方法，即定义常数比热、定义温度和（或）组元相关比热、用分子动力论定义比热。对于燃烧方面的应用，推荐使用温度相关方法定义比热。

比热的定义是在 Materials（材料）面板中完成的。下面将详细介绍每个选项的输入及相关模型。

（1）常数比热。

在 Materials（材料）面板中 c_p 右边的下拉列表中选择 constant（常数）并输入相应的比热值即可。系统默认流体（空气）的比热为 1006.43 J/kg-K。

（2）将比热定义为温度的函数。

可以用阶梯函数、分段多项式和多项式等形式定义比热，具体设置方法请参见 9.1.3 节

中的内容。

（3）用分子动力论定义比热。

在计算气体流动时可以用分子动力论定义比热，其形式如下：

$$c_{p,i}=\frac{1}{2}\frac{R}{M_i}\left(f_i+2\right) \tag{9-37}$$

式中 f_i 是气体组元 i 的能量模式的数量（自由度），R 为普适气体常数，M_i 为组元 i 的分子量，$c_{p,i}$ 为组元 i 的定压比热。在 Materials（材料）面板 c_p 右边的下拉列表中选择 kinetic-theory（分子动力论），求解器就会使用上面的方程来计算比热。

（4）将比热定义为组元浓度的函数

如果流场中包含多种组元，可以将比热定义为组元浓度的函数。定义组元相关比热的步骤为：

1）首先，选择 Cp 右边的下拉列表中的 mixing-law，然后点击 Change/Create（修改/创建）按钮。

2）定义各组元的比热。

3）求解器将用下式计算混合物的比热：

$$c_p=\sum_i Y_i c_{p,i} \tag{9-38}$$

式中 Y_i 为组元 i 的质量浓度，$c_{p,i}$ 为组元 i 的比热。

9.6 辐射特性

在 FLUENT 中使用辐射模型时，除了 S2S 模型外，其他辐射模型都需要定义一些附加属性。这些属性均在 Materials（材料）面板中定义：

（1）对于 P-1 模型，需要设定 Absorption Coefficient（吸收系数）以及 Scattering Coefficient（散射系数），即 P-1 模型方程中方程 1 的 a 和 σ_s。

（2）对于 Rosseland 辐射模型，也需要设定 Absorption Coefficient（吸收系数）以及 Scattering Coefficient（散射系数），即 Rosseland 模型方程中的 a 和 σ_s。

（3）对于 DTRM 模型，只需要设定 Absorption Coefficient（吸收系数），即 DTRM 模型方程中的 a。

（4）对于 DO 模型，需要设定 Absorption Coefficient（吸收系数）以及 Scattering

Coefficient（散射系数），即 DO 模型方程中的 a 和 σ_s。此外，如果模拟半透明介质，需要指定 Refractive Index（折射指数），即 DO 模型的边界条件中的 n_a 或者 n_b。用 DO 模型可以定义固体材料的辐射属性，并在计算半透明介质时予以引用。

9.6.1 吸收系数

可以将吸收系数定义为常值、温度相关函数、组元相关函数或者自定义函数。如果使用 DO 辐射模型的非灰度（non-gray）辐射，还可以选择在每一个灰度带指定常数吸收系数。

吸收系数的单位等于路径长度的倒数。吸收系数与散射系数共同定义了穿过流体介质路径上每个单位长度上的辐射强度的变化。吸收系数可以用 CO_2 和 H_2O 发射率的表格计算。这个表格通常可以在辐射热交换的教科书等相关资料中得到。

（1）常数吸收系数。

只要在 Materials（材料）面板中 Absorption Coefficient（吸收系数）下面的框中输入数值，就可以将吸收系数定义为常数。

（2）将吸收系数定义为组元相关函数。

吸收系数可以定义为组元相关函数，当地 a 值是水蒸气或者二氧化碳的当地质量浓度的函数。这种定义方式可以用于模拟燃烧中的辐射。FLUENT 所使用的可变吸收系数（variable-absorption-coefficient）模型就是模拟燃烧中辐射过程的“灰度气体加权求和模型”（Weighted-Sum-of-Gray-Gases Model, 缩写为 WSGGM）。

在 Materials（材料）面板中 Absorption Coefficient（吸收系数）右边的下拉列表中选择 wsggm-cell-based（基于网格的 WSGGM 模型）、wsggm-domain-based（基于求解域的 WSGGM 模型）或者 wsggm-user-specified（用户自定义的 WSGGM 模型）。这三个 WSGGM 模型的区别在于计算路径长度所使用的方法有所不同。

只有在进行带组元的计算时，才能在列表中看到 wsggm 选项，而且二氧化碳和水必须是混合物的组成成分。

当采用 WSGGM 模型计算吸收系数时，可以选择路径长度的计算方法：

1）如果选择 wsggm-cell-based（基于网格的 WSGGM 模型），则求解器将采用特征网格尺度（characteristic-cell-size）方法进行计算，并且不需要再输入其它参数。

2）如果选择 wsggm-domain-based（基于求解域的 WSGGM 模型），则求解器将采用平均光程（mean-beam-length）法计算 a，而且 FLUENT 将以计算区域的平均尺度为参考值计算平均光程，并且不需要再输入其它参数。

3）如果选择 wsggm-user-specified（用户自定义的 WSGGM 模型），求解器仍然会采用平均光程（mean-beam-length）法进行计算，但是光程是人为定义的，需要在 WSGGM User

Specified（WSGGM 用户设定面板）中的 Path Length（路径长度）框中给定平均光程。在选择 wsggm-user-specified（用户自定义的 WSGGM 模型）选项时，这个面板会自动打开。

（3）非灰度辐射吸收系数的输入。

如果使用非灰度 DO 模型，可以对灰度模型所使用的每一个带（band）指定不同的常数吸收系数。在 Absorption Coefficient（吸收系数）下拉列表中选择 gray-band（灰度带），然后在 Gray-band Absorption Coefficient（灰度带吸收系数）面板中为每一个带定义吸收系数。

（4）烟尘和粒子对吸收系数的影响。

在计算 P-1 和 DO 辐射模型时，如果在 Discrete Phase Model（弥散相模型）面板中激活 Particle Radiation Interaction（粒子辐射相干作用）选项，FLUENT 将在吸收系数的计算中考虑粒子对吸收系数的影响。

如果需要计算烟尘构成，并考虑烟尘对于吸收系数的影响，则在 Soot Model（烟尘模型）面板中打开 Soot Radiation Interaction（烟尘辐射相干作用）下面的 Generalized Model（一般模型）选项。在使用 WSGGM 来计算组元相关吸收系数时，可以在任何一个辐射模型中考虑烟尘的影响。

9.6.2 散射系数

散射系数的缺省值等于 0，并且被假定为各向同性的。可以指定它为常值、温度相关函数（请参阅 9.1.3 节）或者用自定义函数进行定义，还可以将它指定为各向异性相函数。

散射系数的单位为路径长度的倒数。散射系数和吸收系数被共同用来定义通过流体介质路径的每一单位长度上的辐射强度的变化。在计算含有微粒的燃烧问题中，可以考虑采用散射系数。

1．常数散射系数的输入。

在 Materials（材料）面板中的 Scattering Coefficient（散射系数）右边的下拉列表中选择 constant（常数），然后在下面的框中输入数值既完成整个定义过程。

2．散射相函数的输入。

在系统缺省设置中，散射相函数被定义为各向同性函数。FLUENT 中也允许将散射相函数定义为线性各向异性函数。如果采用 DO 模型，那么还可以将散射相函数定义为 Delta-Eddington 和自定义散射函数，下面是这四种函数的设置方法：

（1）在 Scattering Phase Function（散射相函数）下拉列表中选择 isotropic（各向同性），就可以将散射过程设置为各向同性过程。

（2）在 Scattering Phase Function（散射相函数）下拉列表中选择 linear-anisotropic（线性各向异性），然后设定相函数系数，就可以将散射过程设置为线性各向异性过程。

（3）要使用 Delta-Eddington 相函数，请在 Scattering Phase Function（散射相函数）下拉列表中选择 delta-eddington，则 Delta-Eddington Scattering Function（散射相函数）面板将自动打开。在面板中指定 Forward Scattering Factor（前向散射因子）和 Asymmetry Factor（非对称因子）后，设置过程既告结束。

（4）要使用自定义相函数，可在 Scattering Phase Function（散射相函数）下拉列表中选择 user-defined（自定义）相函数。自定义相函数中必须指定前向散射因子和非对称系数。

9.6.3 折射指数

只有在使用 DO 模型模拟半透明介质时才需要设置这个参数。在默认的情况下，折射指数被设置为 1。修改设置的方法很简单，就是在 Refractive Index（折射指数）旁的输入栏中输入折射指数的值即可。

9.7 质量扩散系数

在计算质量扩散过程时可以采用两种模型，一个是 Fick 定律，另一个是全组元扩散模型。大多数情况下可以采用 Fick 定律，只有在扩散过程为主导过程的层流流动中（比如在化学蒸汽的分解过程中）才需要使用全组元扩散模型。

将两种模型进行比较可以发现，全组元扩散模型更完备、准确，但是消耗的系统资源也更多，计算时间更长。

9.7.1Fick 扩散定律

在求解多组元流动输运方程时，需要首先计算质量扩散系数。在 Fick 定律中，质量扩散系数用于计算化学组元的扩散通量：

$$J_i = -\rho D_{i,m} \nabla Y_i - D_{T,i} \frac{\nabla T}{T} \quad \text{（9-39）}$$

式中 $D_{i,m}$ 是混合物中成分 i 的质量扩散系数，$D_{T,i}$ 是热扩散系数，即 Soret 系数。

在混合物组元构成没有变化，或者 $D_{i,m}$ 与组元构成无关时，上述方程严格有效。在混合物中，除了作为载体的气体组元外，其他气体成分的质量浓度远远小于 1 时，混合物可以作为稀释气体处理，Fick 定律近似成立；在混合物为非稀释气体时，FLUENT 用多组元

扩散模型进行计算。FLUENT 允许以多种方法指定 $D_{i,m}$，包括指定 $D_{i,j}$（成分 i 在 j 中的二元质量扩散系数），但是并不直接使用 $D_{i,j}$，而是将二元扩散系数 $D_{i,j}$ 与混合物扩散系数 $D_{i,m}$ 通过下式联系起来，在计算中直接采用混合物质量扩散系数 $D_{i,m}$ 进行计算：

$$D_{i,m} = \frac{1 - X_i}{\sum_{j,j \neq i} X_j / D_{ij}} \qquad (9\text{-}40)$$

式中 $D_{i,j}$ 时成分 i 在 j 中的二元质量扩散系数，X_i 是成分 i 的摩尔浓度。每个化学组元都有与之对应的 $D_{i,m}$ 和 $D_{i,j}$，计算中需要逐个进行输入。

在湍流流动中，层流流动的 Fick 扩散定律的方程由下式替代：

$$J_i = -\left(\rho D_{i,m} + \frac{\mu_t}{Sc_t} \right) \nabla Y_i - D_{T,i} \frac{\nabla T}{T} \qquad (9\text{-}41)$$

式中 Sc_t 是湍流的有效 Schmidt 数：

$$Sc_t = \frac{\mu_t}{\rho D_t} \qquad (9\text{-}42)$$

D_t 是由湍流引起的有效质量扩散系数。

进行了湍流修正的 Fick 定律保留了层流 Fick 定律中的各项，而用涡粘度系数 μ_t 与 Schmidt 数的商代表湍流对质量扩散过程的影响。层流质量扩散主要是由分子间运动引起的，而湍流则是通过宏观运动实现的，因此湍流对质量扩散的影响要远远大于层流。在这种情况下，层流的影响几乎可以忽略不计，所以在输入各项参数时，可以不必修改系统的缺省设置，将层流质量扩散系数 $D_{i,m}$ 设置为一个常数。

在层流计算中，还可以用全组元扩散模型计算质量扩散通量。在全组元扩散模型计算中，质量扩散通量是通过求解 Maxwell-Stefan 方程得到的。

Fick 定律中涉及到的热扩散系数 $D_{T,i}$ 则可以用常数、多项式、用户自定义函数和系统自带的经验公式计算得出。

9.7.2 设置质量扩散系数

质量扩散系数的设置有两种方式：一种是直接设置 $D_{i,m}$；另一种是先设置 $D_{i,j}$，再用公式（9-7-2）算出 $D_{i,m}$。$D_{i,m}$ 的设置方法又分为两种：一种是将所有组元的 $D_{i,m}$ 设置为同一个常数，另一种则是将各组元的 $D_{i,m}$ 设置成不同的常数，或用多项式定义为温度的函数。在采用全组元扩散模型时，需要逐个设置各组元的二元扩散系数 $D_{i,j}$。

在稀疏混合物中，一般使用直接设置 $D_{i,m}$ 的方法；在非稀释混合物中，一般使用先设置 $D_{i,j}$，再计算 $D_{i,m}$ 的方法。

在采用全组元扩散模型时，则只能逐个设置各组元的 $D_{i,j}$。激活全组元扩散模型的方法是在 Species Model（组元模型）面板中打开 Full Multicomponents Diffusion（全组元扩散）选项，然后在 Materials（材料）面板中选择全组元扩散方法。

$D_{i,m}$ 和 $D_{i,j}$ 都是在 Materials（材料）面板中进行输入的，所采用的单位是 m^2/s（国际单位制），或 ft^2/s（英制）。

如果要将所有 $D_{i,m}$ 定义为同一个常数，可以在 Mass Diffusivity（质量扩散）旁的下拉列表中选择 constant-dilute-appx（常数稀释混合物近似），然后在 $D_{i,m}$ 输入栏中输入一个常数即可。

如果要分别设置各组元的 $D_{i,m}$，可以在下拉列表中选择 dilute-appx（稀释混合物近似），然后在随即打开的 Mass Diffusion Coefficients（质量扩散系数）面板（如图 9-13）中进行设置。设置过程为，首先在 Species Di（组元 Di）列表框中选定一个组元，然后在下面的

Coefficient（系数）下面选择一种定义方式（常数、多项式），最后按照 9.1.3 节中的方法完成相关定义即可。

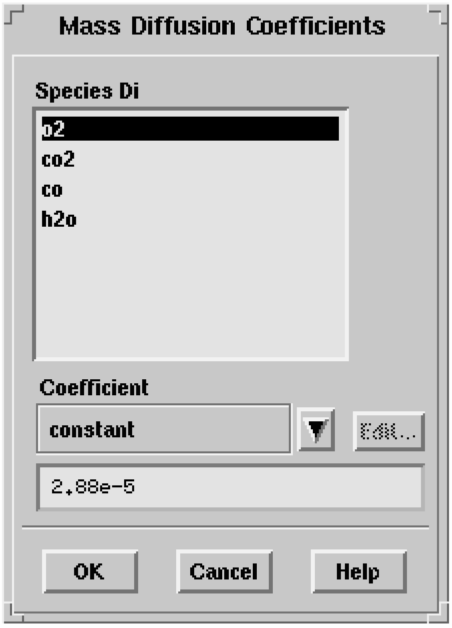


图 9-13 Mass Diffusion Coefficients（质量扩散系数）面板

如果要定义 $D_{i,j}$，则在 Mass Diffusivity（质量扩散）旁的下拉列表中选择 multicomponent（多组元），然后在随即打开的 Mass Diffusion Coefficients（质量扩散系数）面板（如图 9-14）中，成对地选择组元 *i* 和组元 *j*，然后在下面的 Coefficients（系数）下面选择定义方法（常数、多项式），最后按照 9.1.3 节中介绍的方法完成定义即可。

热扩散系数 $D_{T,i}$ 的定义方法包括 kinetic-theory（分子运动论）、specified（专门定义）和 user-defined（用户自定义）三种形式。如果采用分子运动论，则无需输入任何参数，系统内部会自动选择相关公式进行计算。如果采用 Specified（专门定义）方式，则可以使用包括常数、多项式等定义方式，具体方法请见 9.1.3 相关内容，这里就不再详细介绍了。用户自定义方法见后面相关章节中的介绍。

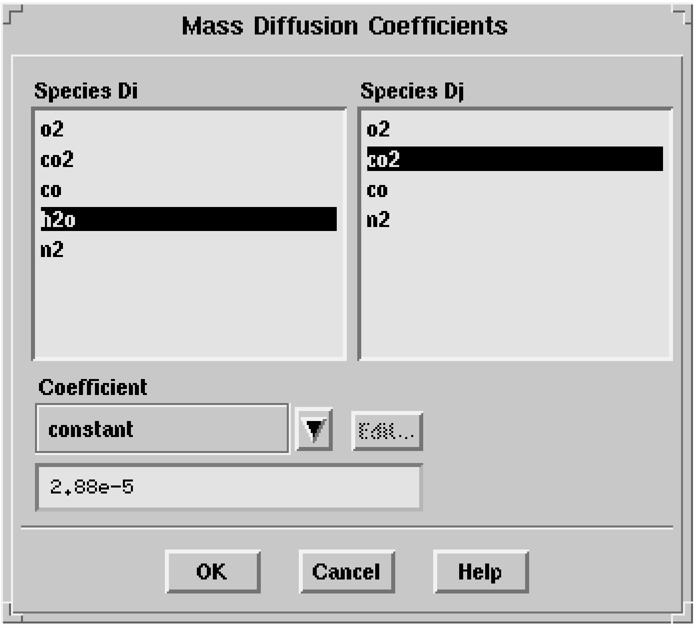


图 9-14 Mass Diffusion Coefficients（质量扩散系数）面板

9.8 其他物性参数

9.8.1 标准状态焓

如果采用有限速率模型或者涡扩散模型计算反应流动，则需要定义每个组元的标准状态焓（也叫做生成焓或者生成热）用于定义混合物的焓。

标准状态焓和参考温度在 Materials（材料）面板中的 Standard State Enthalpy（标准状态焓）和 Reference Temperature（参考温度）中输入。

9.8.2 标准状态熵

如果有限速率模型中包含逆向反应，则需要为每个组元定义标准状态熵，用于计算混合物的熵。

标准状态熵和参考温度在 Materials（材料）面板中的 Standard State Entropy（标准状态

熵）和 Reference Temperature（参考温度）中定义。

9.8.3 分子热传导系数

计算预混燃烧时需要设定未燃烧混合物的材料性质，包括分子热传导系数α。α的定义为：

$$\alpha = \frac{k}{\rho c_p} \qquad (9\text{-}43)$$

式中k为热传导系数，ρ为密度，c_p为定压比热。

α在标准条件下的数值可以在燃烧手册中查到。非标准条件下的数值，则需要利用第三方具有完整化学反应模型的一维燃烧程序计算得出。

分子热传导系数在 Materials（材料）面板中的 Molecular Heat Transfer Coefficient（分子热传导系数）栏中设定。

9.8.4 分子运动论参数

在使用理想气体假设时，可以用分子运动论定义粘性、热传导系数、比热和质量扩散系数。

在用分子运动论计算流体粘性时，需要输入分子运动论参数σ和ε / k_B。这些参数是 Lennard-Jones 参数，在 FLUENT 中分别被称为 characteristic length（特征长度）和 energy parameter（能量参数）。在分子运动论用于计算流体的热传导系数时不用输入任何参数。如果使用分子运动论计算比热，则需要输入流体分子运动的自由度。如果使用分子运动论定义混合物的质量扩散系数，则需要输入每个组元的σ_i和$(\varepsilon / k_B)_i$。

在选用 kinetic-theory（分子运动论）作为上述参数的定义方法后，需要输入的参数包括 L-J Characteristic Length（L-J 特征长度）、L-J Energy Parameter（L-J 能量参数）和 Degree of Freedom（分子自由度）。

特征长度的单位是埃，能量参数的单位是绝对温度单位，自由度是无量纲数。在默认情况下，所有的分子运动论参数都设为零，不同材料的具体数值可以查阅相关文献。

9.9 操作压强

对于不同的流动状态，操作压强对计算的意义也不尽相同。本节将介绍在不同情况下如何正确设定操作压强。

9.9.1 低马赫数流动中截断误差对压力计算的影响

在低马赫数可压流动中，总的压降和绝对静压相比很小，因此数值截断会对其有很大的影响。其原因在于：当马赫数 M 远远小于 1 时，压降 Δp 与动压头 $\frac{1}{2}\gamma p M^2$ 相关，式中 p 为静压，γ 为比热比。由此可以看出 Δp 与 M^2 的关系，即当 M -> 0 时 Δp -> 0。因此，除非给予足够的注意，否则低马赫数流动计算很容易受到截断误差的影响。

9.9.2 操作压强、表压和绝对压力

操作压强（operating pressure）、表压（gauge pressure）和绝对压强（absolute pressure）是让初学者容易感到头痛的概念。使用者的困惑主要来源于三者之间的相互关系，下面就对此做个简单介绍。

前面已经讲过截断误差对压降 Δp 计算的影响，为了减小截断误差的不利影响，FLUENT 引入了表压这个概念。所谓表压，就是绝对压强与操作压强的差，而操作压强则大致等于流场的平均压强。因为在计算过程中实际使用的是表压，而压强增量与表压相比已经不再是一个小量，也就不会被计算误差淹没，所以可以大大降低截断误差对计算结果的影响。

式（9-44）为绝对压强 p_{abs}、操作压强 p_{op} 和表压 p_{gauge} 之间的关系式——绝对压强是操作压强和表压之和：

$$p_{abs} = p_{op} + p_{gauge} \tag{9-44}$$

在 FLUENT 计算和显示过程中用到的压强都是表压。

9.9.3 操作压强的设定

1．操作压强的意义

操作压强对于不可压理想气体流动和低马赫数可压流动来说是十分重要的，因为不可压理想气体的密度是用操作压强通过状态方程直接计算出来的（参见公式（9-8）)，而在低马赫数可压流动中操作压强则起到了避免截断误差的负面影响的重要作用。

对于高马赫数可压缩流动，操作压强的意义不大。在这种情况下，压力的变化比低马赫数可压流动中压力的变化大得多，因此截断误差不会产生什么影响，也就不需要使用表压进行计算。事实上，在这种计算中使用绝对压力会更方便。因为 FLUENT 总是使用表压进行计算，所以需要在这类问题的计算中将操作压强设置为零，而使表压和绝对压力相等。

如果密度假定为常数，或者密度是从温度的型函数中推导出来，那么根本就不使用操作压强。操作压强的缺省值为 101325 Pa.。

2．操作压强的设定

如前所述，如何设置操作压强取决于马赫数的变化范围和密度的定义方式。例如，在不可压理想气体的计算中，应该将操作压强的值设为流场中压强的平均值。表 9-1 是设定操作压强的推荐方法。

表 9-1 操作压强推荐值

密度定义方式	马赫数范围	操作压强
理想气体定律	M>0.1	0 或者流场平均压强
	M<0.1	流场平均压强
温度型函数	不可压缩流动	不用设置
常值	不可压缩流动	不用设置
不可压理想气体	不可压缩流动	流场平均压强

操作压强在 Operating Pressure（操作压强）面板中设定。启动面板的菜单操作顺序为：Define/Operating Conditions...

9.9.4 参考压力位置

对于不包括压力边界的不可压缩流动，FLUENT 会在每次迭代之后调整表压场来避免数值漂移。每次调整都要用到（或接近）参考压力点网格单元中的压强，即在表压场中减去单元内的压力值得到新的压力场，并且保证参考压力点的表压为零。如果计算中包含了压力边界，上述调整就没有必要了，求解过程中也就不再用到参考压力位置。

参考压力位置被缺省设置为原点或者最接近原点的网格中心。如果要改变参考压强位置，比如将它定位在压强已知的点上，则可以在 Operating Pressure（操作压强）面板中输入参考压强位置的新的坐标值（X，Y，Z）。

9.10 真实气体模型

气体在高压下会偏离理想气体假设，因此用理想气体模型进行计算就会产生误差。在这种情况下需要采用真实气体模型进行计算。Fluent 软件中的耦合求解器提供两种真实气体模型用于这种计算：

（1）NIST 真实气体模型。

（2）用户自定义真实气体模型。

9.10.1 NIST 真实气体模型

1．NIST 真实气体模型简介

NIST 是 National Institute of Standards and Technology（美国国家标准与技术学会）的简写。NIST 真实气体模型利用 NIST 提供的制冷剂与制冷剂混合物的热力学和输运性质数据库第 6 版（Refrigerants and Refrigerants Mixtures Database Version 6, REFPROP v6.0）中的数据，计算 33 种纯流体的热力学性质和输运性质。在选择使用真实气体模型后，FLUENT 会自动调用 REFPROP v6.0 中的数据用于计算流体的热力学性质和输运性质，而且该数据库一旦被调用，则 Materials（材料）面板中所显示的属性值全部失效，后处理过程中报告和显示的也都是真实气体模型的热力学性质和输运性质。

NIST 真实气体模型的应用有如下限制：

（1）只能在耦合求解器中使用。

（2）Materials（材料）面板上的数据失效，而且不能用 Materials（材料）面板修改物性参数。

（3）REFPROP 数据库中的数据不能被改动。

（4）在使用 NIST 真实气体模型时，流场的各区域中的所有气体都是真实气体。

（5）NIST 模型中不支持混合物和饱和气体。

（6）被假定为真实气体的气体必须是纯气体、单一组元、单相、过热蒸汽。

2．NIST 真实气体模型的使用

使用 NIST 真实气体模型时，除 Materials（材料）面板外，其它软件功能与使用理想

气体模型时相同。使用步骤如下：

（1）在 Solver（求解器）面板中选择耦合求解器。

（2）用文本命令激活 NIST 模型，即在 FLUENT 主窗体中键入文字命令：

```
> define/user-defined/real-gas/nist-real-gas-model
```

在系统出现下面的提示时，回答 yes：

```
use real gas? [no]   yes
```

窗口中随即出现可用气体名称列表：

```
CO2.fld     R123.fld     R142b.fld    R236fa.fld   butane.fld
R11.fld     R124.fld     R143a.fld    R245ca.fld   ethane.fld
R113.fld    R125.fld     R152a.fld    R245fa.fld   isobutan.fld
R114.fld    R13.fld      R22.fld      R32.fld      propane.fld
R115.fld    R134a.fld    R227ea.fld   R41.fld      propylen.fld
R116.fld    R14.fld      R23.fld      RC318.fld
R12.fld     R141b.fld    R236ea.fld   ammonia.fld
```

从上面的列表中选择一种气体，例如“R125.fld”，即在下面的提示后面键入气体名称：

```
select real-gas data file [""] "R125.fld"
```

系统随即从数据库中加载这种气体的相关数据，并在窗口中报告相关信息：

```
/usr/local/Fluent.Inc/fluent6.1/realgas/lib/R125.fld

Opening "/usr/local/Fluent.Inc/fluent6.1/realgas/
ultra/librealgas.so"...
Setting material "air" to a real-gas...

matl name: "R125"
           : "pentafluoroethane"
           : "354-33-6"
Mol Wt     : 120.022
Critical properties:
  Temperature : 339.33 (K)
  Pressure    : 3.629e+06 (Pa)
  Density     : 4.75996 (mol/L) 571.3 (kg/m^3)
```

（3）在设置完成后保存算例文件（.cas 文件）。

（4）开始计算并对计算结果进行后处理。

9.10.2 用户自定义真实气体模型

用户自己建立的真实气体模型也只能在耦合求解器下使用。用户自定义模型只能用 C 语言编写，且必须遵循一定的规则。用户自定义的真实气体模型可以仿照样例文件进行建立。在模型编译完毕后，相关数据将被保存在共享库文件中，并可以在计算中被调用。

建立用户自定义真实气体模型（UDRGM）的步骤如下：

（1）定义真实气体状态方程和所有相关的热力学和输运性质方程。

（2）编写符合格式要求的 C 程序。

（3）启动 FLUENT 并建立算例文件。

（4）编译 C 程序并建立共享库文件。

（5）顺序键入下列文字命令，调用共享库文件：

define -> user-defined -> real-gas text command.

（6）开始计算。

需要说明的是，该模型假定真实气体为各向同性、可压缩物质。

1．书写 C 函数库。

函数库的书写比较简单，但必须使用特定的函数名称和宏名称，基本要求如下：

（1）在程序开始处必须包含 udf.h 头函数，从而允许在编译过程中使用 DEFINE 宏和 FLUENT 中的子函数。

（2）就象下面的例子中那样，程序中必须包含至少一个 UDF 中的 DEFINE 函数（比如 DEFINE_ON_DEMANGD），以便可以使用编译过的 UDF 函数模块。

（3）程序中各种变量的单位为国标单位。

（4）在函数库中必须使用下面主函数集合中的函数。这些函数用来将函数库中的热力学数据传送到求解器中。其中大写的 ANYNAME 可以被任意改写为用户自己定义的函数名。程序中用到的变量包括温度（T）、压力（P）、密度（ρ）和组元质量浓度（Y_i[]）。

下面是列出这些函数，并给予简短说明：

1）void ANYNAME_error（int err, char *f, char *msg），用于打印错误信息。

2）void ANYNAME_Setup（Domain *domain, char *filename, int（*messagefunc）（char *format, ...）, void（*errorfunc）（char *format, ...）)，用于建立模型和初始化。与所建立的模型相关的数据和参数都可以用这个函数读入。

3）double ANYNAME_density（double T, double P, double yi[]），返回作为温度与压力函数的密度值。此函数为计算中所用到的状态方程。

4）double ANYNAME_specific_heat（double T, double Rho, double yi[]），返回作为温度和密度的函数的定压比热。

5）double ANYNAME_enthalpy（double T, double Rho, double yi[]），返回作为温度和密度函数的焓。

6）double ANYNAME_entropy（double T, double Rho, double yi[]），返回作为温度和密度函数的熵。

7）double ANYNAME_mw（double yi[]），返回分子量。

8）double ANYNAME_speed_of_sound（double T, double Rho, double yi[]），返回作为温度和密度函数的声速。

9）double ANYNAME_viscosity（double T, double Rho, double yi[]），返回作为温度和密度函数的动力粘度。

10）double ANYNAME_thermal_conductivity（double T, double Rho, double yi[]），返回作为温度和密度函数的热传导系数。

11）double ANYNAME_rho_t（double T, double Rho, double yi[]），返回常压下作为温度和密度函数的 $\frac{d\rho}{dT}$ 的值。

12）double ANYNAME_rho_p（double T, double Rho, double yi[]），返回常压下作为温度和密度函数的 $\frac{d\rho}{dP}$ 的值。

13）double ANYNAME_enthalpy_t（double T, double Rho, double yi[]），返回常压下作为温度和密度函数的 $\frac{dh}{dT}$ 的值。

14）double ANYNAME_enthalpy_p（double T, double Rho, double yi[]），返回常温下作为温度和密度函数的 $\frac{dh}{dp}$ 的值。

在程序结束时，必须定义 RGAS 数据结构，结构中的成员为指向上述主函数的指针，属于 RGAS_Function 类型，名称为 RealGasFunctionList。

注意：函数指针必须遵循下面的顺序：

```
UDF_EXPORT   RGAS_Functions RealGasFunctionList =
{
  ANYNAME_Setup,                    /* Setup initialize   */
  ANYNAME_density,                  /* density   */
  ANYNAME_enthalpy,                 /* enthalpy   */
  ANYNAME_entropy,                  /* entropy   */
  ANYNAME_specific_heat,            /* specific_heat   */
```

```
    ANYNAME_mw,                        /* molecular_weight    */
    ANYNAME_speed_of_sound,            /* speed_of_sound    */
    ANYNAME_viscosity,                 /* viscosity    */
    ANYNAME_thermal_conductivity,      /* thermal_conductivity    */
    ANYNAME_rho_t,                     /* drho/dT |const p    */
    ANYNAME_rho_p,                     /* drho/dp |const T    */
    ANYNAME_enthalpy_t,                /* dh/dT |const p    */
    ANYNAME_enthalpy_p                 /* dh/dp |const T    */
};
```

2．编译 C 函数并建立共享库文件。

编译 C 函数需要使用 C 编译器。UNIX 系统中自带了 C 编译器，在 WINDOWS 系统中需要安装 C＋＋编译器（VC＋＋6.0 或更高版本）以便进行 C 函数的编译。

首先通过编译 C 函数建立 UDRGM 库，并将所建立的 UDRGM 库载入 FLUENT。编译 C 函数的步骤如下：

（1）把 C 程序放在工作目录下。

（2）启动 FLUENT，读入算例文件（.cas 文件）。

（3）在图形界面下，或在文字命令界面下编译 C 程序并建立共享库文件。在编译成功后，共享库文件将被放在 ntx86/2d 目录中，文件名为 libudf.dll（WINDOWS 系统），或 libudf.so（UNIX 系统）。

图形界面的编译请参考 UDF 手册。

文字界面下的编译过程为：

1）选择菜单项：

define -> user-defined -> compiled-functions .

2）选择编译选项。

3）输入被编译的库文件名（不包括后缀）。

4）输入 C 程序源文件名。

5）输入头文件名，如果没有额外的头文件，直接按回车键跳过这步操作。

上述操作结束后，FLUENT 就开始编译过程，并将编译所产生的共享库文件放入相应的目录。

下面看一个例子，粗体字母为用户输入的文字命令或操作动作，其余为系统显示信息：

>_define/user-defined/compiled-functions

    load OR compile ? [load]> **compile**

    Compiled UDF library name: ["libudf"]   **my_lib**

在这里需要确认 UDF 源文件与算例文件和数据文件在同一个目录下。如果已经存在一

个 libudf 目录，请将其删除以保证系统调用的是最新生成的文件。

Continue?[yes]　　<回车>

Give C-Source file names:

First file name: [""]　　**my_c_file.c** <回车>

Next　file name: [""]　　<回车>

Give header file names:

First file name: [""]　　**my_header_file.h** <回车>

3. 载入 UDRGM 共享库文件。

载入过程用文字命令完成，具体步骤为：

（1）键入下列命令进入相关目录：

define -> user-defined -> real-gas

（2）选择 user-defined-real-gas-model 目录。

（3）打开真实气体模型：

use real gas? [no]　　yes

系统随即询问共享库文件的位置，只要键入共享库文件所在目录的名称或者完整的路径，系统就可以找到库文件。载入成功后，系统会显示与下面信息相类似的信息：

Opening user-defined realgas library "RealgasLibraryname"...

Library "RealgasDirName/lnx86/2d/libudf.so" opened

Setting material "air" to a real-gas...

Loading Real-RealGasPrefexLable Library:

4. UDRGM 程序例子：真实气体状态方程。

这里用一个算例演示 UDRGM 源程序的格式。在这个例子中使用的变量包括压强（p）、温度（T）、比热（c_p）、焓（H）、熵（S）、密度（ρ）、音速（c）和普适气体常数（R）。用到的方程包括理想气体状态方程：

$$\rho = \frac{p}{RT} \tag{9-45}$$

将比热设置为常数 $c_p = 1006.42$，则焓的定义为：

$$H = c_p T \tag{9-46}$$

熵的定义为：

$$S = c_p \log\left(\frac{T}{T_{ref}}\right) + R\log\left(\frac{p_{ref}}{p}\right) \tag{9-47}$$

带下标 ref 的变量为参考物理量，其中 T_{ref}=288.15 K，p_{ref}=101325 Pa。音速定义为：

$$T = \sqrt{Tc_p \frac{R}{c_p - R}} \tag{9-48}$$

密度导数为：

$$\left(\frac{d\rho}{dp}\right)_T = \frac{1}{RT} \tag{9-49}$$

$$\left(\frac{d\rho}{dT}\right)_p = -\frac{p}{RT^2} = -\frac{\rho}{T} \tag{9-50}$$

焓导数为：

$$\left(\frac{dH}{dT}\right)_p = c_p \tag{9-51}$$

$$\left(\frac{dH}{dp}\right)_T = \frac{c_p}{\rho R}\left(1 - \frac{p}{\rho}\frac{d\rho}{dp}\right) = 0 \tag{9-52}$$

与上述模型对应的 C 源程序如下：

```
/**********************************************************************/
/* User Defined Real Gas Model :                                      */
/* For Ideal Gas Equation of State                                    */
/*                                                                    */
/**********************************************************************/
#include "udf.h"
#include "stdio.h"
#include "ctype.h"
#include "stdarg.h"

#define MW 28.966          /* molec. wt. for single gas (Kg/Kmol) */
#define RGAS (UNIVERSAL_GAS_CONSTANT/MW)
```

```
static int    (*usersMessage)(char *,...);
static void (*usersError)(char *,...);


DEFINE_ON_DEMAND(I_do_nothing)
{
   /* This is a dummy function to allow us to use */
   /* the Compiled UDFs utility                    */
}



void IDEAL_error(int err, char *f, char *msg)
{
   if (err)
      usersError("IDEAL_error (%d) from function: %s\n%s\n",err,f,msg);
}

void IDEAL_Setup(Domain *domain, char *filename,
                 int    (*messagefunc)(char *format, ...),
                 void (*errorfunc)(char *format, ...))
{
   /* Use this function for any initialization or model setups*/
   usersMessage = messagefunc;
   usersError     = errorfunc;
   usersMessage("\nLoading Real-Ideal Library: %s\n", filename);
}

double IDEAL_density(double Temp, double press, double yi[])
{
   double r = press/(RGAS*Temp); /* Density at Temp & press */

   return r;                     /* (Kg/m^3) */
```

```
}

double IDEAL_specific_heat(double Temp, double density, double yi[])
{
   double cp=1006.43;

   return cp;                                /* (J/Kg/K) */
}

double IDEAL_enthalpy(double Temp, double density, double yi[])
{
   double h=Temp*IDEAL_specific_heat(Temp, density, yi);

   return h;                                 /* (J/Kg) */
}

#define TDatum 288.15
#define PDatum 1.01325e5
double IDEAL_entropy(double Temp, double density, double yi[])
{
   double s=IDEAL_specific_heat(Temp,density,yi)*log(Temp/TDatum)+
                                  RGAS*log(PDatum/(RGAS*Temp*density));
   return s;                                 /* (J/Kg/K) */
}

double IDEAL_mw(double yi[])
{
   return MW;                                /* (Kg/Kmol) */
}

double IDEAL_speed_of_sound(double Temp, double density, double yi[])
{
   double cp=IDEAL_specific_heat(Temp,density,yi);
```

```
    return sqrt(Temp*cp*RGAS/(cp-RGAS));    /* m/s */
}

double IDEAL_viscosity(double Temp, double density, double yi[])
{
    double mu=1.7894e-05;

    return mu;                                  /* (Kg/m/s) */
}

double IDEAL_thermal_conductivity(double Temp, double density,
                                                        double yi[])
{
    double ktc=0.0242;

    return ktc;                                 /* W/m/K */
}

double IDEAL_rho_t(double Temp, double density, double yi[])
{
    /* derivative of rho wrt. Temp at constant p */
    double rho_t=-density/Temp;

    return rho_t;                               /* (Kg/m^3/K) */
}
double IDEAL_rho_p(double Temp, double density, double yi[])
{
    /* derivative of rho wrt. pressure at constant T */
    double rho_p=1.0/(RGAS*Temp);

    return rho_p;                               /* (Kg/m^3/Pa) */
}

double IDEAL_enthalpy_t(double Temp, double density, double yi[])
```

```
{
   /* derivative of enthalpy wrt. Temp at constant p */
   return IDEAL_specific_heat(Temp, density, yi);
}

double IDEAL_enthalpy_p(double Temp, double density, double yi[])
{
   /* derivative of enthalpy wrt. pressure    at constant T      */
   /* general form dh/dp|T = (1/rho)*[ 1 + (T/rho)*drho/dT|p] */
   /* but for ideal gas dh/dp = 0                                */
   return 0.0 ;
}

UDF_EXPORT RGAS_Functions RealGasFunctionList =
{
   IDEAL_Setup,                    /* initialize           */
   IDEAL_density,                  /* density              */
   IDEAL_enthalpy,                 /* enthalpy             */
   IDEAL_entropy,                  /* entropy              */
   IDEAL_specific_heat,            /* specific_heat        */
   IDEAL_mw,                       /* molecular_weight     */
   IDEAL_speed_of_sound,           /* speed_of_sound       */
   IDEAL_viscosity,                /* viscosity            */
   IDEAL_thermal_conductivity,     /* thermal_conductivity */
   IDEAL_rho_t,                    /* drho/dT |const p     */
   IDEAL_rho_p,                    /* drho/dp |const T     */
   IDEAL_enthalpy_t,               /* dh/dT |const p       */
   IDEAL_enthalpy_p                /* dh/dp |const T       */
};
/**************************************************************/
```

仿照这个程序可以建立自己的共享库文件。

第十章 移动与变形区域流动计算

在 FLUENT6.1 中，移动与变形区域流动的计算功能得到了极大的加强。FLUENT 在计算中可以使用移动坐标系、移动网格区域计算移动和变形区域中的流动。移动网格区域的运动被定义为与之相连的坐标系的运动。通过 FLUENT 提供的计算模型，可以模拟物体在流场中运动时流场的变化。FLUENT 提供了多种移动单元模型，如何选择取决于运动的复杂程度、涉及的流动机理等，其中最常用的模型便是动网格模型。

10.1 移动区域模拟方法概述

FLUENT 可以进行整个计算区域或者部分区域存在移动的流动模拟，包括单个旋转坐标系和多旋转坐标系、平移坐标系的计算。单旋转坐标系选项适合于旋转机械、搅拌器以及其他相关设备的模拟。由于设备中的转子、推进器、叶片周期性的运动，这些设备下的流动都是惯性条件下的非定常流动。但在没有定子的情况下，流动相对于旋转部件来说，就变成了定常流动，这样流动的分析就可以大大得到简化。在存在定子的情况下，我们只能采用相对复杂，能够进行转子或者推进器计算的模型。共有四种模型可供选择：

1．多参考系（MRF）模型。
2．混合面模型。
3．滑动网格模型。
4．动态网格技术。

多参考系模型和混合面模型都假定流动为定常，转子或者推进器的影响可以用近似平均值来代替。这种处理方式在转子与定子间的相互影响较弱时可以得到较好的结果。而滑动网格模型，则假定流动是非定常的，因此可以真实地模拟转子与定子间的相互影响，所以在两者相互影响不可忽略的情况下，应当选择滑动网格模型。当然选择该模型需要消耗更多的计算时间。

10.2 旋转坐标系中的流场计算

10.2.1 简介

虽然 FLUENT 中的所有计算都是在惯性系中进行的，但是 FLUENT 也有计算非惯性

系中流场的能力。非惯性系流场的描述方程中包含了对坐标系加速度的描述。旋转设备中的流动便是一个非常典型的例子。这种径向加速度为常值的旋转流动可以用坐标系随旋转设备运动的方式来进行计算。在 FLUENT 软件中可以使用旋转坐标系计算这种流场。图 10-1 是这种流动的一个典型示例，图中也说明了旋转坐标系与固定坐标系在使用上的区别。

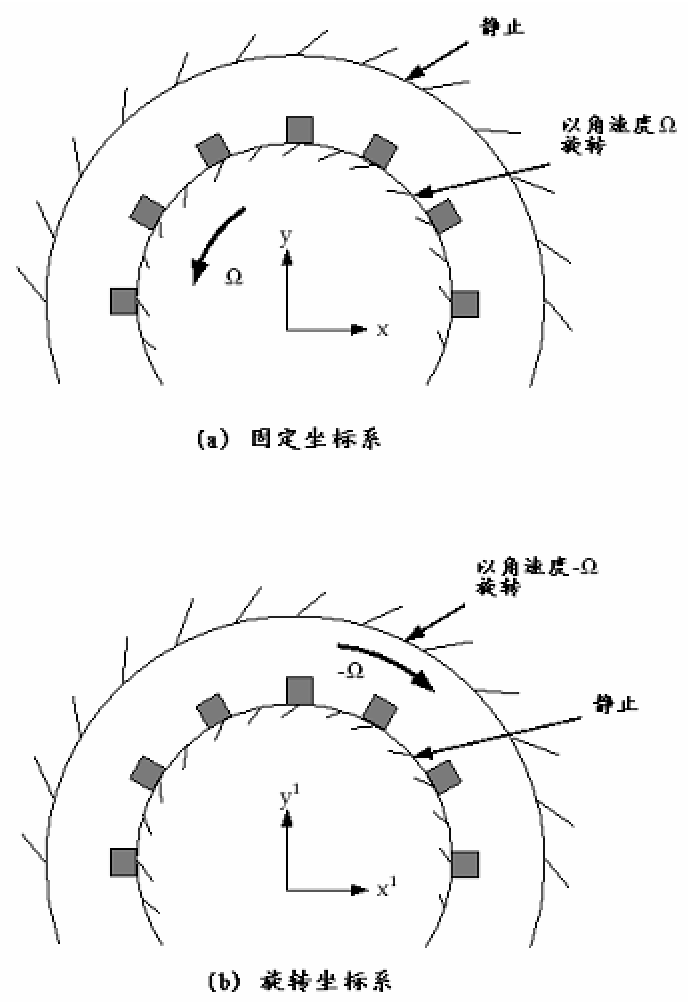


图 10-1 变固定坐标系为旋转坐标系

1. 旋转坐标系的应用

图 10-2 中显示了可以应用旋转坐标系解决的几种问题，包括搅拌器中叶轮的旋转，旋转机械中叶片的旋转以及旋转通路中的流动计算等。应用该模型可以把旋转边界变化为固定边界以方便计算。

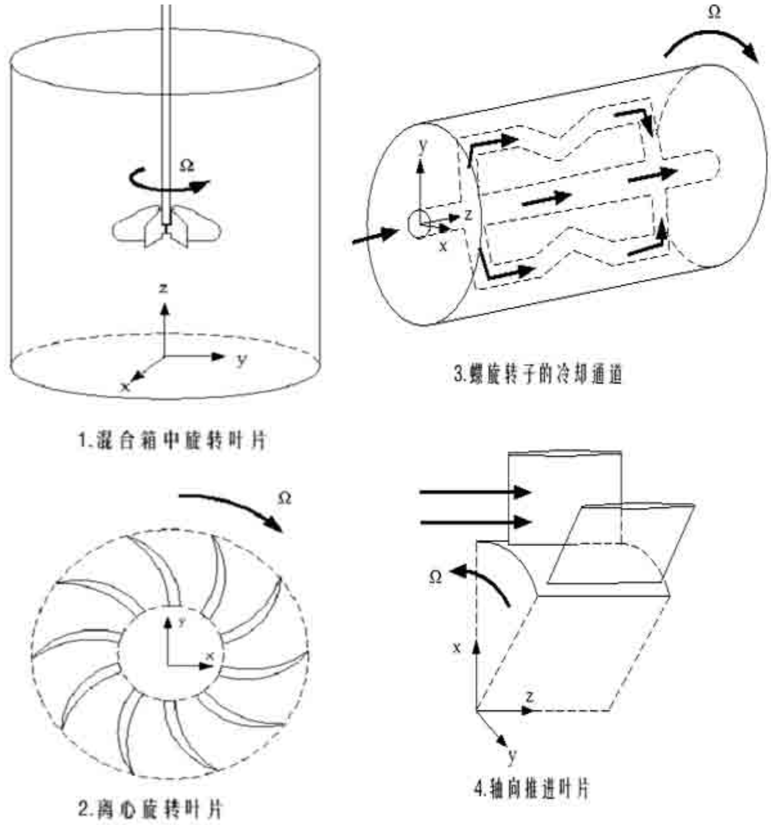
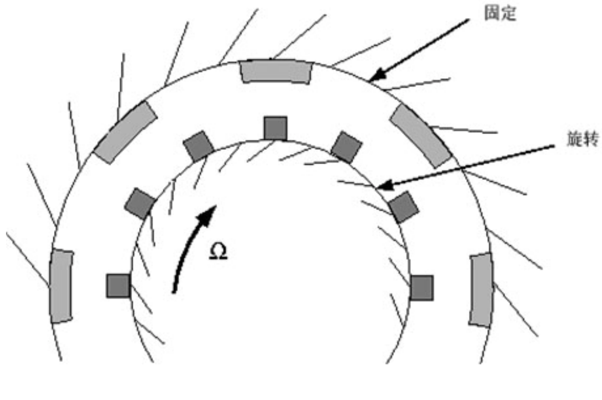


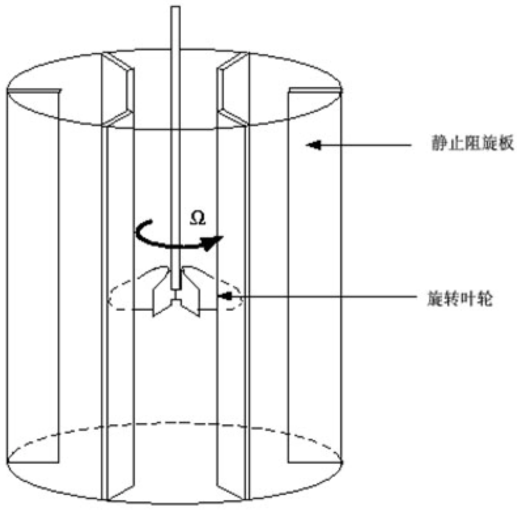
图 10-2 旋转坐标系应用示例

2. 模拟转子-定子相互作用

由于转子定子相互作用问题不能通过简单的坐标转换而得到解决，所以 FLUENT 提供了多重参考系（MRF）模型、混合面模型和滑动网格模型来计算这类问题。这些方法在下面的章节中都有详细的讲解。



1. 转子与定子的相互作用



2. 装备阻旋板中的叶轮

图 10-3 需要 MRF、混合面和滑动网格模型的算例

10.2.2 旋转坐标系方程

在旋转坐标系下求解质量守恒以及连续性方程时，在动量方程中需要添加流体的加速度项。在 FLUENT 中求解旋转坐标系中的问题时可以使用两种速度，绝对速度 $\vec{v}$ 与相对速度 $\overrightarrow{v_r}$。两者的关系如下式：

$$\vec{v}_r = \vec{v} - \left(\vec{\Omega} \times \vec{r}\right) \qquad (10\text{-}1)$$

式中 $\vec{\Omega}$ 为旋转角速度，$\vec{r}$ 为旋转坐标系下的位置向量。

惯性坐标系下动量方程的左侧为：

$$\frac{\partial}{\partial t}(\rho\vec{v})+\nabla\cdot(\rho\vec{v}\vec{v}) \qquad (10\text{-}2)$$

在旋转坐标系下，动量公式的左侧用绝对速度可以写成下式：

$$\frac{\partial}{\partial t}(\rho\vec{v})+\nabla\cdot(\rho\vec{v}_r\vec{v})+\rho(\vec{\Omega}\times\vec{v}) \qquad (10\text{-}3)$$

用相对速度可以写为：

$$\frac{\partial}{\partial t}(\rho\vec{v}_r)+\nabla\cdot(\rho\vec{v}_r\vec{v}_r)+\rho(2\vec{\Omega}\times\vec{v}_r+\vec{\Omega}\times\vec{\Omega}\times\vec{r})+\rho\frac{\partial\vec{\Omega}}{\partial t}\times\vec{r} \qquad (10\text{-}4)$$

式中 $\rho(2\vec{\Omega}\times\vec{v_r})$ 为哥氏力。在 FLUENT 计算中，忽略了（10-4）中的最后一项，所以利用相对速度公式不能准确模拟角速度随时间变化的流动。

旋转域中的质量守恒方程或者连续性方程可以参考下式写成绝对速度或相对速度的形式：

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\vec{v}_r)=S_m \qquad (10\text{-}5)$$

10.2.3 单旋转坐标系网格的建立

在用旋转坐标系统解决问题时，必须注意如下的限制：

（1） 对于二维问题，旋转轴必须平行于 Z 轴。

（2） 对于二维轴对称问题，旋转轴必须为 x 轴。

（3） 对于三位问题，旋转区域的网格必须在确定旋转轴的情况下进行划分。通常为方便起见，采用坐标轴，X、Y、Z 轴，做旋转轴。但是 FLUENT 中允许使用任意方向的直线做旋转轴。

10.2.4 单旋转坐标系中计算模型的设置

在求解单旋转坐标系问题时，必须完成下列输入：

（1） 在 Solver（求解器）面板中选择计算中采用的速度形式，可以是 Absolute（绝对速度）或者 Relative（相对速度）。

（2） 对于流场中的每个网格单元区域设定参考系的旋转角速度（Ω）和旋转轴，具体步骤为：

首先，打开 Define（定义）菜单下的 boundary conditions（边界条件）面板，然后在

Fluid（流体）面板或者 Solid（固体）面板中指定 Rotate-Axis Direction（旋转轴方向）和 Rotate-Axis Origin（旋转轴的原点）。然后，在面板中的 Motion type（运动类型）下拉菜单中选择 Moving Reference Frame（运动参考系）选项，并在 Rotational Speed（旋转速度）下的 Speed（速度）中参考系的运动速度。

（3） 定义壁面的速度，可以是相对速度或者绝对速度。在定义壁面的速度时极其容易出错误，请千万小心。例如，对于静止坐标下的静止壁面在旋转坐标系中要定义该壁面的速度为绝对参考坐标下的零值。这样在后来需要修改旋转速度时就不需重新设定壁面的速度。

（4）定义速度入口的速度与压强入口的压强和流动方向。对于速度入口，有相对速度与绝对速度两种选择。对于压强入口，压强和流动方向的定义都可以是绝对的或者相对的，这取决于第一步所选择的速度公式。

10.2.5 相对速度与绝对速度的选择

绝对速度适用于流动区域中的大部分区域不旋转的流动，而相对速度适用于大多数区域都旋转的情况。在大多数情况下，两者并无本质区别，但在耦合求解器中只可使用绝对速度公式。

对于速度入口和壁面边界，可以任意将速度设定为绝对或者相对速度，而不用理会计算中所用的是相对还是绝对速度，但压强边界的设定则没有这么自由，必须遵循一定的限制。简单说来就是一致性原则，如果速度为绝对速度，则压强和流动方向也必须为绝对坐标系下的值；如果速度为相对速度，则压强与流向也必须在相对参考系中设定。

对于压强出口边界，指定的静压与参考系无关。但是如果有回流，指定的静压值会被认作是总压。对于应用绝对速度的计算，指定的静压被认为等同于绝对坐标系下的总压；对于应用相对速度的计算，指定的静压被认为等同于参考系下的总压。在出现反向流动的边界上，如果速度用绝对速度形式定义，则 FLUENT 假定绝对速度垂直于压强出口；而在采用相对速度形式时，则假定相对速度垂直于压强出口。

10.2.6 旋转坐标系求解策略

求解旋转坐标系问题与求解轴对称问题有非常相似的一个难点，那就是旋转项影响很大时，动量方程之间的高度耦合关系。当旋转速度较大时，径向压强梯度会同时影响轴向和径向流动，从而在流场中产生旋转与螺旋运动。由于高度耦合的模型在计算中造成求解失稳，所以必须采用一些特殊的手段来避免不稳定的产生并获得收敛解，可以采用的技术包括：

（1）转换速度公式的参考坐标体系，该技术只对分离求解器有效。

（2）用 PRESTO!格式求解，此种技术只对分离求解器有效。

（3）在压强与螺旋速度梯度较大的地方，确保网格细化得当。

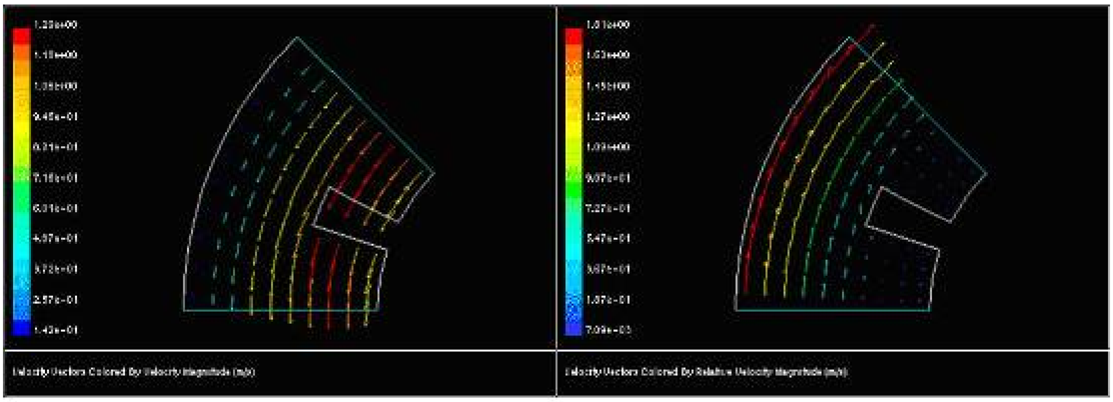
（4）在采用分离求解器时，将速度亚松弛因子减小到 0.3 到 0.5，或者更低。

（5）采用逐步增加旋转速度的方法，先使用小的旋转速度，然后逐步提高速度直到获得收敛解。该方法为提高旋转坐标系下计算稳定性的有效手段。比如可以在计算开始时用最大旋转速度的 10%做第一次计算。若计算成功，再用最大旋转速度的 20%进行计算，依此类推，逐渐增加旋转速度，直到达到最大旋转速度。

10.2.7 单旋转坐标系的后处理

求解完成后，要对计算结果进行后处理。这里补充一点个人看法：一个好的数据处理过程有时比计算过程更费精力也更重要，因为只有通过后处理产生的图表数据，别人才能直观、系统地看清你工作的意义！对于单旋转坐标系下的计算数据，我们可以处理两种速度参数，即绝对速度参数和相对速度参数。对于总压、总温，以及任何取决于参考系的参数，都有相对、绝对两种参数可供选择。

处理数据时应注意，图形的着色可以是相对参数值，也可以是绝对参数值。两者的图形往往差别很大，如图 10-4。



绝对速度着色　　相对速度着色

图 10-4 相对速度与绝对速度着色对比

10.3 MRF 模型

MRF 模型是一种定常计算模型，模型中假定网格单元做匀速运动，是本章介绍的四种方法中最简单的一种方法。这种方法适用于网格区域边界上各点的相对运动基本相同的问题。大多数时均流动都可以用 MRF 模型进行计算，特别是运动网格区域与静止网格区域间的相互作用比较微弱时可以使用 MRF 模型进行计算，例如搅拌器内流场计算、泵和风机内流场计算等等。MRF 模型的另一个用途是用来为滑动网格模型计算提供初始流场，即先用 MRF 模型粗略算出初始流场，再用滑动网格模型完成整个计算。

为了更好地说明 MRF 模型的应用，下面给出一个实际应用的例子。

1. 算例一：叶轮计算

对于只有一个叶轮的搅拌器，可以在包含叶轮及其周围流体的网格区域上定义一个旋转坐标系，然后在叶轮区域以外的网格区域上定义一个静止坐标系。在计算中采用定常计算方式，即假定流动为定常流动，两个网格区域在交界面上的流体速度是相同的。

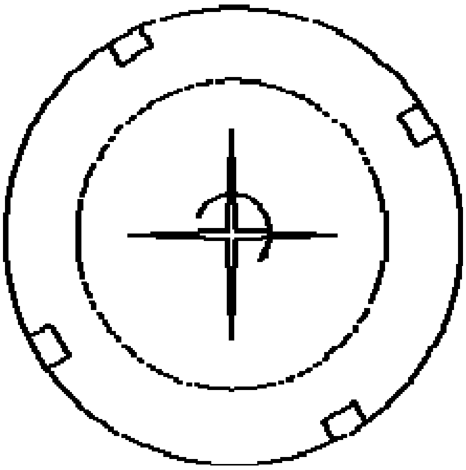


图 10-5 单叶轮水箱示意图

也可以计算包含多个旋转坐标系的问题。图 10-6 中为一个包含两个相邻叶轮的水箱，要解决这个问题需要三个参考坐标系：固壁上取静止坐标系，每个叶轮取一个旋转坐标系。

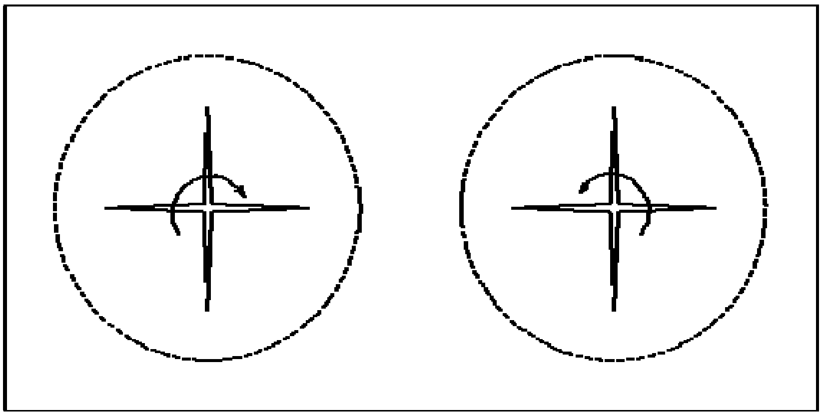


图 10-6 两个叶轮水箱

2. 计算中的限制

建议不要在 Realizable $k-\varepsilon$ 模型中使用 MRF 模型。

在选择分割运动区域与邻近区域的边界面时，必须使垂直于边界的速度等于零。

一般来说，MRF 方法只适用于定常流动。在 FLUENT 中也可以用它来计算非定常流动，区别在于此时的输运方程中都会加入非定常项。不过与其这样做，不如用滑动网格来计算。

FLUENT 用相对速度计算颗粒轨迹和迹线。对于无质量颗粒，所得迹线是以流体流线为基准，用相对速度计算出来的。对于有质量颗粒，在后处理中显示的迹线是没有意义的。与 MRF 模型耦合计算的弥散相模型也没有意义。

为了解决颗粒追踪和耦合弥散相计算的问题，可以考虑用绝对速度代替相对速度。可以用文本命令 define/models/dpm/tracking/track-in-absolute-frame 将速度形式从相对速度转换为绝对速度。

在 MRF 模型中使用相对速度将不能准确模拟轴对称旋转流动，因为 FLUENT 软件目前无法将速度转换为旋转速度。另外，在使用 MRF 模型时，FLUENT 假定平移与旋转速度都是常值，因此在网格区域边界定义中不能使用随时间变化的速度。

3. 相对速度与绝对速度的区别

MRF 模型因所用速度的不同而会有很大差别。在采用相对公式进行计算时，如果遇到跨越两个相邻网格子域的计算，比如计算网格子域边界上的扩散速度，则 FLUENT 会将两个子域的速度首先转化为绝对速度，然后再进行相关计算，最后将结果重新转换为相对速度形式返回各子域的计算。若采用绝对速度，则每个子区域的控制方程中的速度为绝对速度。所以相邻区域间的速度不需要特别的转化。

4. MRF 模型网格设置

有两种网格设置方式：

（1）若相邻不同参考系间的边界网格是正则的，即相邻子域在边界上共用一组节点，那么可以如往常一样建立网格，并将所有网格分区保存在同一个网格文件中。流场中采用了不同参考坐标系的区域需要相应地为它设置独立的网格分区，同时应该将各分区的交界面设置为 interior（内部）类型。

（2）若相邻边界网格是非正则的，即各子域在边界上分别有一组相互不重合的网格节点，则要应用非正则网格划分方法，在 Grid Interfaces（网格交界面）面板中将相邻的边界面定义为 interface（交界面）。Grid Interfaces（网格交界面）面板可以用下述菜单操作打开：

Define -> Grid Interfaces...

详细设置参见网格划分相关章节。

5. MRF 模型的设置和求解

应用 MRF 模型时，除了一般性的设置，还需要完成下面一些特殊的设置：

（1）在 Solver（求解器）面板中，选择 Relative（相对速度）或 Absolute（绝对速度）。

（2）为每个网格区域指定平移速度和（或）旋转角速度。

若被设置区域做旋转运动，则在 Boundary Conditions（边界条件）面板中启动 Fluid（流体）或者 Solid（固体）面板，并在面板中指定 Rotate-Axis Origin（旋转轴原点）和 Rotate-Axis Direction（旋转轴方向）。

同样在 Fluid（流体）或 Solid（固体）面板中，无论区域是平动还是转动，都要在 Moving Type（运动类型）下拉列表中将参考系指定为 Moving Reference Frame（运动参考系），并在 Rotational Speed（旋转速度）下设定 Speed（速度），或在 Translational Speed（平动速度）速度下设定平动速度的 X、Y、Z 分量。

（3）定义壁面边界条件的速度条件。通常将壁面与坐标系进行绑定设置，即如果壁面与坐标系一起运动，或同时保持静止，则都可以将壁面速度设为“0”。这样设置可以在将来需要改变流体区域的运动速度时，不用再改变壁面边界条件的设置，而只要改变流体运动速度就可以了。

（4） 定义速度入口速度以及压强入口总压和流向。对于速度入口，可以用绝对速度或相对于相邻网格区域的相对速度定义速度边界条件。在压强边界条件中，总压和流动方向的定义要根据第（1）步中定义的速度形式来决定，即总压和流动方向的值取决于前面选择了相对速度形式还是绝对速度形式。

MRF 模型的求解过程和求解策略没有特殊之处，这里不再赘述。

10.4 混合面模型

10.4.1 简介

混合面模型是介于 MRF 模型与滑动网格之间的一种求解一个或多个区域有相对运动的模型。

MRF 模型适用于邻近区域在交界面上的流动接近均匀的情况，如果流动不均匀，可以选用滑动网格模型，但是在某些情况下采用滑动网格模型可能使计算复杂化，例如多级旋转机械问题，这时便可以考虑采用混合面模型。

在混合面模型中，每一个流动区域被当作定常流场计算。在混合面各相邻区域的交界面上，邻近区域的流场数据是作为边界条件进入其他区域的。混合面模型忽略了周围环境的随机扰动对流场的影响，这对时间平均流场的计算不会有太大的影响。

使用混合面模型必须满足下列要求：

（1）必须使用绝对速度进行计算。

（2）不能在 LES（大涡模拟模型）中使用混合面模型。

（3）组元输运和燃烧计算不能与混合面模型同时使用。

（4）通用多相流模型（VOF、混合物、欧拉模型）不能与混合面模型同时使用。

（5）不能与弥散相模型同时用于耦合求解器中，而只能在分离求解器中使用。

10.4.2 混合面模型理论

1. 转子与定子区域

考虑如图 10-7、图 10-8 所示的涡轮机械。图 10-7 是一个半径为常数的面，而图 10-8 中为混合设备中角度θ为常数的面。在这两种情况中，都包含两个流动区域，即旋转的转子区域与静止的定子区域。

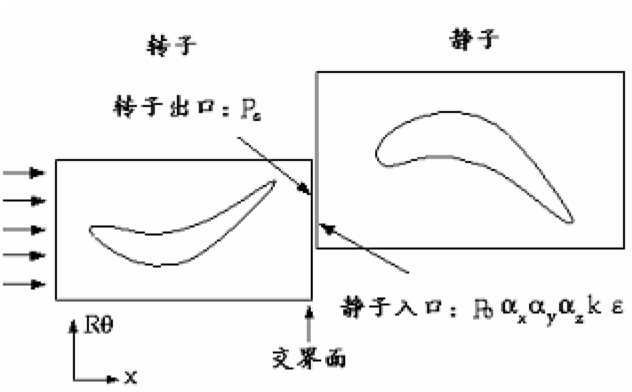


图 10-7 轴向转子定子相互影响

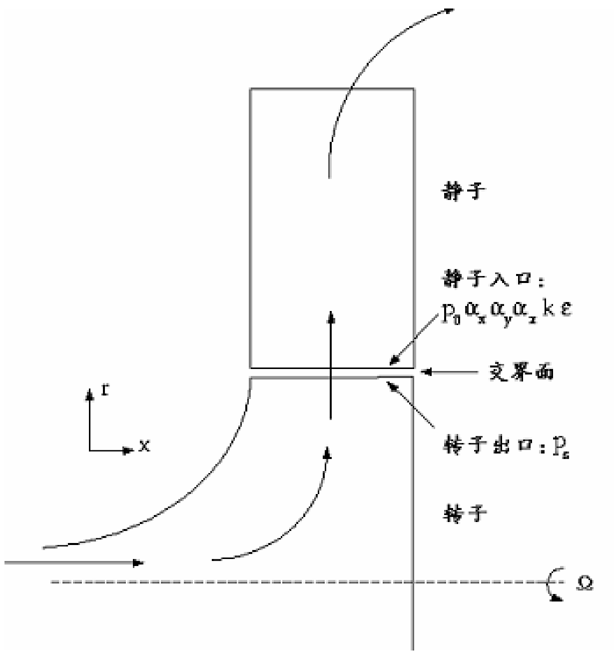


图 10-8 径向转子定子相互影响

在计算中，每个区域采用单独的网格划分。在相邻的网格区域之间，利用混合面模型进行耦合。可以利用混合面模型对任意多个网格区域进行计算。图中的转子区域与定子区域的网格划分可以是非正则的，就是转子与定子交界面上的节点不一定要完全匹配。

2. 混合面概念

混合面的中心思想是把非定常流场划分为若干个定常的流动区域进行计算。在计算中，FLUENT 用面积加权平均法对转子区域与定子区域交界面上的流场变量沿周向进行平均，然后用平均得到的变量分布作为下一个迭代步中交界面上的边界条件。在如图 10-7 与图 10-8 所示例子中，转子出口总压（p_0）、当地流动方向的方向余弦（$\alpha_r,\alpha_t,\alpha_z$）、湍流动能 k、湍流耗散率 ε 等参数的平均值，在下个迭代步中被用作定子入口的边界条件。类似地，定子入口的静压值、方向余弦（$\alpha_r,\alpha_t,\alpha_z$），在新的迭代步中被用作转子出口的边界条件。采用这种数据传递方式的前提是在交界面上采用特定的边界条件类型。上游的出口与下游的入口的耦合被称作“混合面对”。为了在 FLUENT 中产生混合面对，上下游边界条件必须满足如下配合关系：

表 10-1 混合面边界设定规则

上游	下游
压强出口	压强入口
压强出口	速度入口
压强出口	质量入口

3. FLUENT 混合面算法

混合面算法的基本步骤如下:

（1）进行转子区域与定子区域的流场计算。

（2）计算转子出口、定子入口流场变量的平均值，并将结果分别作为下个迭代步中对方的边界条件，从而完成两个区域间的数据信息传递。

（3）将平均值设定为边界条件。

（4）重复（1）至（3）步的计算过程，直到计算收敛。

4. 质量守恒关系

如果在上面介绍的计算步骤中使用压强入口和压强出口条件，则在混合面前后质量不是严格守恒的。如果采用质量入口和压强出口条件，则 FLUENT 将在混合面前后强制满足质量守恒关系。计算过程为首先计算上游区域（压强出口）的质量流量，并据此修正质量入口的质量流量，从而可以保证质量流量在混合面前后严格守恒。这种修正在每个迭代步中都会进行，从而保证在整个计算过程中质量流量的守恒。

5. 扭矩守恒

在系统缺省设置下，FLUENT 不能在混合面前后保证扭矩守恒。对于扭矩转换器，总扭矩应该为零，因此强制扭矩在混合面前后守恒非常有必要的，在 FLUENT 中也是一个可能的选择，否则周向动量的累积会造成周向平面上出现扭矩的源或汇，而这显然是不符合实际情况的。因为与扭矩相关是切向速度，而与质量流量相关的是轴向速度，所以可以同时使用扭矩守恒与质量守恒关系，二者之间没有冲突。

6. 总焓守恒

在系统缺省设置下，FLUENT 不要求混合面前后的总焓满足守恒关系。但是在需要计算推进效率的问题中，需要保证穿过混合面的总焓满足守恒关系。总焓守恒在 FLUENT 中总焓守恒是模型设置中的一个选项，因此是可以做到的。

让总焓守恒的方法，简单说来就是通过修正下游总温分布，从而使下游的总焓能与上游的总焓相匹配。在每个迭代步中都进行这种修正，就可以保证在整个计算过程中总焓保持守恒。

10.4.3 混合面模型的设置

1. 问题建立步骤:

混合面模型设置的基本步骤如下：

（1）在 Solver（求解器）面板中选择绝对速度选项。

（2）设定每个网格区域的角速度及旋转轴，具体方法与 MRF 模型中相同。

（3）在边界条件中定义边界面上的速度。

（4）定义速度入口条件中的速度，压强入口条件和质量流入口条件中的总压和流动方向。速度入口条件中的速度可以用绝对速度和相对速度两种方式定义，而压强入口条件和质量流入口条件中的总压和流动方向只能用绝对速度进行计算。在质量流入口条件中不需要定义质量流量或质量通量，在设定混合面模型后，系统会自动激活 Mass Flux with Averaged Mass Flux（平均质量通量）选项并设定相关参数的值。如前所述，混合面的出口必须使用压强出口条件。

（5）在 Mixing Plane（混合面）面板中定义混合面，启动该面板的菜单操作如下：

Define -> Mixing Planes

然后指定组成混合面的两个平面区域，在 Upstream Zone（上游区域）和 Downstream Zone（下游区域）中选择构成混合面区域的两个面。

在三维计算中，再在 Mixing Plane Geometry（混合面几何特征）中指定混合面的方向。选项 Radial（径向）表示所计算的问题与图 10-7 中的算例类似，所有变量是半径 *r* 的函数，这个选项适用于轴向流动机械。选项 Axial（轴向）表示所计算的问题与图 10-8 中的算例

类似，所有变量都简化为轴向坐标 *x* 的函数，这个选项适用于径向流动机械。

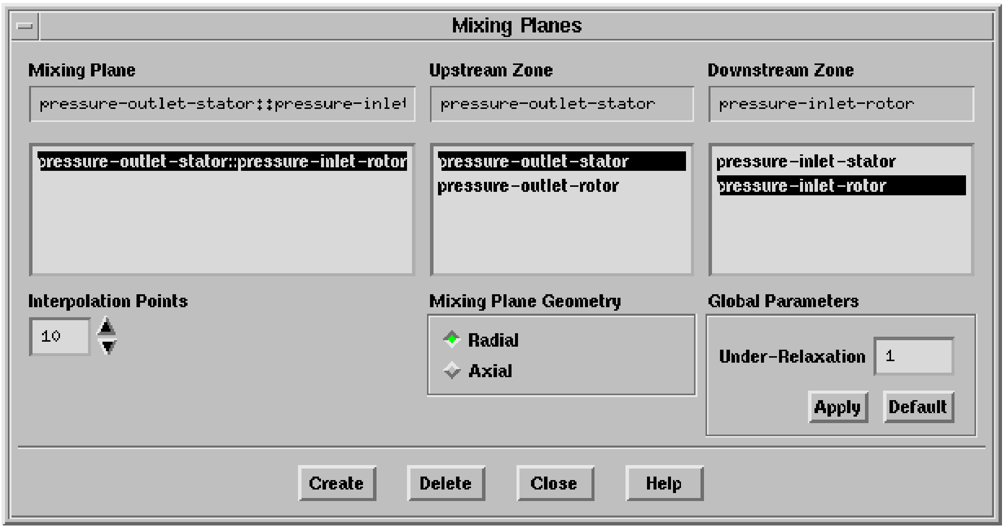


图 10-9 Mixing Planes（混合面）面板

在三维计算中，设定插值点（Interpolation Points）数量，这些插值点就是用来做周向平均的插值点。为了提高周向平均值的计算精度，这个点数应当接近于表面网格的节点数。

另一个需要设置的参数是 Globle Parameters（全局参数）下面的 Under-relaxation（亚松弛因子），在开始计算的时候将这个参数设置为小于 1 的数，可以避免计算在初始流场的剧烈调整过程中发散，其原理与其他计算中的亚松弛因子相同，这里不再赘述。

在设置完毕后，点击 Create（创建）按钮完成混合面的创建。

2. 混合面模型中的选项

混合面模型有两个选项，一个是应用于不可压流计算的固定压强选项，一个是前面介绍的扭矩守恒选项。

对于某些旋转机械，例如扭矩转换器，不存在压强保持不变的边界，因而在用混合面模型进行计算时，可能无法获得收敛解。要解决这个问题，FLUENT 中提供了强制固定压强水平的选项。在应用此选项时，FLUENT 将在每次求解后用参考压强点（Reference Pressure Location）附近的压强作为基准，自动修正流场内各点的表压值。

激活该选项的文字命令如下：

define -> mixing-planes -> set -> fix-pressure-level

激活扭矩守恒选项的文字命令如下：

define -> mixing-planes -> set -> conserve-swirl

然后键入 enable?，并回答 yes，就可以激活扭矩守恒选项。

激活总焓守恒选项的文字命令如下：

define -> mixing-planes -> set -> conserve-total-enthalpy

然后键入 enable?，并回答 yes，就可以激活总焓守恒选项。

10.4.4 混合面模型求解策略和后处理

如果交界面附近没有回流，则利用混合面模型可以取得理想的结果。但是在刚开始计算的时候，原本没有回流的区域也可能出现回流，此时应用混合面模型可能无法获得收敛解。解决办法有两个，一个是在刚开始计算时，在交界面附近采用固定的边界条件，以便获得一个接近真实情况的初始流场；另一个是减小松弛因子，在计算稳定后，再逐步增大亚松弛因子。

混合面模型的计算结果可以用绝对速度和相对速度两种方式进行后处理，可以计算和显示的参数包括速度、马赫数、总压和总温等等。

10.5 滑动网格模型

在需要掌握转子与定子间相互作用随时间变化的细节时必须使用滑动网格模型。如果仅仅希望得到时间平均数据，则可以使用上面介绍的 MRF 模型或者混合面模型。

滑动网格模型不仅可以模拟旋转机械中的呈周期性变化的非定常流动，也可以模拟非周期性的流动，例如隧道中两辆汽车的迎面驶过时周围流场的变化。

1. 滑动网格技术

在滑动网格模型计算中，流场中至少存在两个网格区域，每一个区域都必须有一个网格界面与其他区域之间连接在一起。网格区域之间沿界面做相对运动。

在选取网格界面时，必须保证界面两侧都是流体区域。在计算如图 10-10 中的转子与定子流场时，界面必须取在转子与定子之间，而不能取在转子或定子的一条边上。

在计算过程中，网格区域间沿界面发生相对滑动。图 10-10 与图 10-11 是滑动前后的网格位置对比。

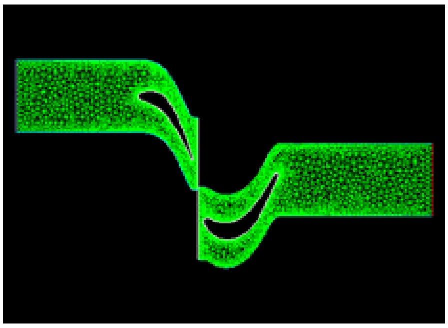


图 10-10 网格初始位置

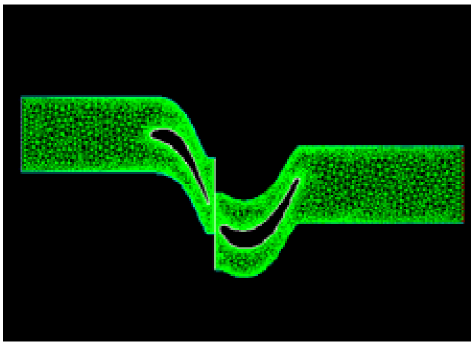


图 10-11 网格变化后位置

在网格区域做相对运动的过程中，不要求两个网格区域在界面上的节点相互重合。因为流动是非定常的，所以必须用非定常格式进行求解。

2. 网格界面形状

从几何角度看，网格界面为相邻两个网格区域所共有，因此两个网格区域在共同边界上的网格边界面具有相同的几何形状，但是网格界面具体是什么形状没有限制，可以是平面、圆柱面、圆锥面或扇形区域，只要能够保证两个网格间在运动过程中彼此不发生重叠即可。

10.5.2 滑动网格理论

滑动网格模型允许相邻网格间发生相对运动，而且网格界面上的节点无需对齐，即网格交界面是非正则的。在使用滑动网格模型时，计算网格界面上的通量需要考虑到相邻网格间的相对运动，以及由运动形成的重叠区域的变化过程。

在实际的计算过程中，每迭代一次就需要重新确定一次网格界面的重叠区域。如果流场是周期性流场，比如转子和定子流场，则两个网格界面相互重合部分形成的区域被称为内部区域，即两侧均为流体的区域，而不重合的部分则被称为周期区域。如果流场是非周期性的，比如汽车交错行驶的流场，则网格界面重合部分仍然被称为内部区域，而不重合的部分则被称为“壁面”区域，如图 10-12 所示。流场变量穿过界面的通量是用内部区域计算的，而不是用交界面上的网格计算的。

看一个例子：在图 10-13 所示的例子中，界面区域由面 A-B、B-C、D-E 和面 E-F 构成。交界区域可以分为 a-d、d-b、b-e 等等。处于两个区域重合部分的面为 d-b、b-e 和 e-c，这三个面共同构成内部区域，其他的面（a-d、c-f）则为成对的周期性区域。要计算穿过区域 IV 的流量，所用的面不是面 D-E，而是用面 d-b 和面 b-e 面，分别计算从 I 和 III 流入 IV 的流量。

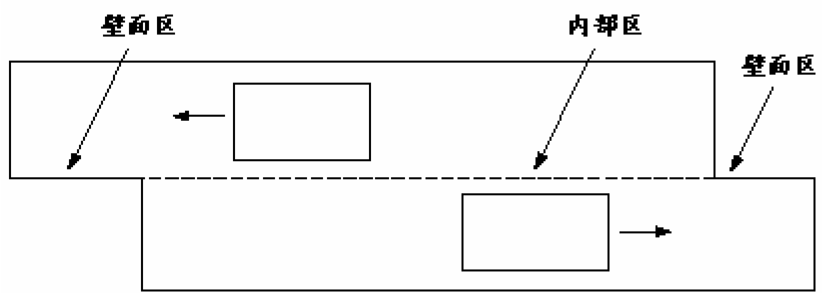


图 10-12 非周期性界面重合产生的区域

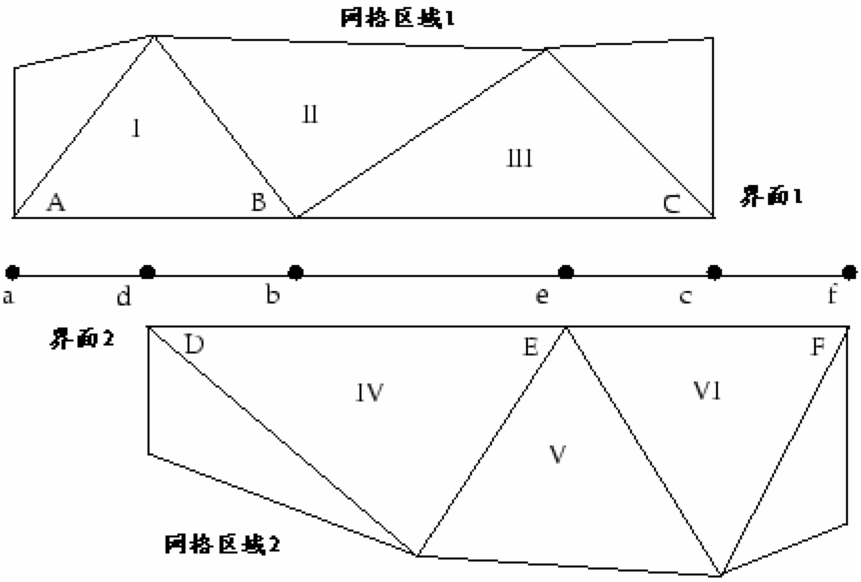


图 10-13 二维网格界面

10.5.3 滑动网格模型的建立与求解

1. 网格要求

滑动网格模型所用网格必须满足下列要求：

（1）以不同速度运动的区域由不同的网格区域构成。

（2）相邻网格区域间不存在垂直于网格界面的运动。

（3）网格界面可以是任何形状，但是交界面的几何形状必须相同。

（4）如果一个网格中存在多个区域，则必须确认在滑动边界上单独设置了一个面区域。交界面两侧的面区域应具有相同的位置和形状，但是每个面区域分别属于各自的网格区域。

（5）在计算转子定子这类周期性问题时，转子网格的周期角和定子网格的周期角必须相等。

（6）必须明确指定周期性区域的运动形式（转动或平动）。

（7）对于三维问题，如果界面是周期性的，则只交界面两侧只能存在一对周期性区域。

2. 计算模型设置

滑动网格模型设置步骤如下：

（1）在 Solver（求解器）面板中选择非定常流动格式。

（2）在 Boundary Conditions（边界条件）面板中定义滑动边界条件：

首先将交界面定义为 interface（交界面），然后在 Fluid（流体）面板或 Solid（固体）面板中，从 Moving Type（移动方式）中选择 moving mesh（移动网格），并设定各自速度。

只有在平动方向与转动轴方向重合时才能同时模拟平动和转动。

在默认情况下，壁面边界速度为零。因此在网格参考系中，可以将壁面条件设为无滑移条件，即壁面随网格一起运动。因此除非壁面在绝对坐标系中静止，就是说在网格参考系中存在相对运动，否则无需修改壁面边界的速度设置。

（3）在 Grid Interfaces（网格界面）面板中定义网格交界面，启动这个面板的菜单操作为：

Define -> Grid Interfaces...

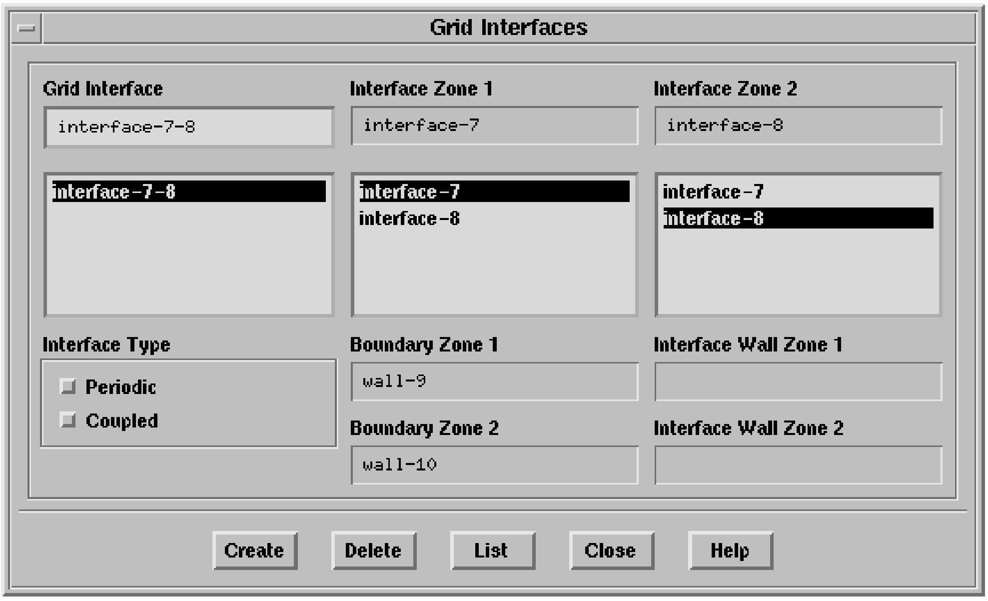


图 10-14 Grid Interfaces（网格界面）面板

首先在 Grid Interface（网格交界面）栏中输入交界面名称，然后指定交界面区域两侧的网格面区域 Interface Zone 1（交界面 1）和 Interface Zone 2（交界面 2）。再指定界面类型为 Periodic（周期型）或 Coupled（耦合型）。最后点击 Create（创建）按钮，完成交界面创建工作。

3. 问题求解

类似于一般非定常问题求解，首先初始化流场，然后按指定时间步数与步长开始计算。在每个迭代步之间，网格都会按照指定的平移速度或旋转速度变化，即重新计算确定界面区域的重合部分。

4. 保存 Case 和 data 文件

用 FLUENT 的自动保存功能将每步计算的结果都保存起来，在后处理过程中就可以用这些数据连续观察流场的变化过程。

注意，必须每一步都保存 case 和 data 数据，因为网格位置每一步都有变化，要想读入 data 文件必须有相应的网格文件。

对于要分几个阶段进行计算的问题，可以采用计算、保存数据、关闭 FLUENT、重新启动 FLUENT、重新开始计算的流程完成整个计算。为了避免网格的变形，可以在保存 case 文件和 data 文件之前，删掉网格界面（grid interface），在重新开始计算时，可以重新读入文件，并重新创建新的网格界面。

5. 时间周期性问题的求解

在求解周期性流场时，如果 T 是流场变化周期，那么流场变量 Φ 的变化规律可以表示为：

$$\Phi(t) = \Phi(t + NT) \qquad (N = 1,2,3,\cdots) \tag{10-6}$$

即每隔一个固定的周期 T，流场变量就循环变化一次。对于转子定子问题，旋转周期就等于叶片间夹角与角速度的商，即 $\mathrm{T} = \theta / \Omega$，其中 θ 为叶片间夹角，Ω 为转子角速度。对于二维转子定子问题 $T=P/V_b$，式中 P 叶片间距离，V_b 是叶片速度。每个周期中的时间步数可以通过把时间周期除以时间步长的办法来确定。如果两个周期之间流场变量的变化小于 5%，则可以认定计算已收敛。

要判定计算是否收敛，需要比较同一点在两个周期中的数值。例如，若时间周期是 5 秒，可以在计算经历过初始阶段（假设为 20 秒）的振荡后，比较 30 秒的结果和 35 秒的结果，看二者相差多少。然后再比较 35 秒与 45 秒时的差别，这样一个周期一个周期地判断下去，直到两个周期间计算结果的差距小于 5%为止。用于判断收敛的变量可以是一些全局变量，例如升力和阻力系数，或质量流量等等，图 10-15 是升力系数随时间的变化。

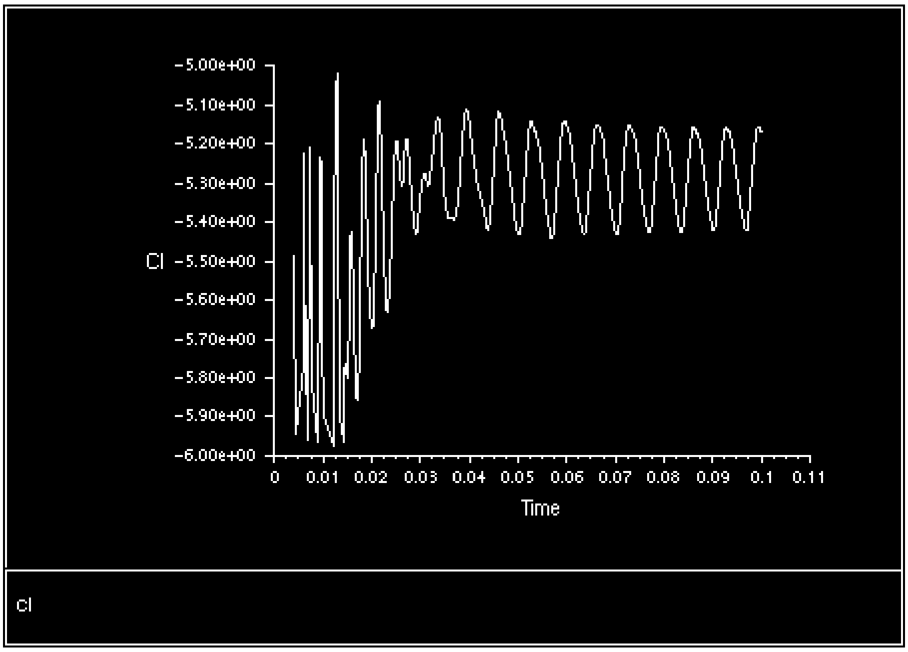


图 10-15 时间周期流场中升力系数的变化

计算初始阶段的振荡与真实的物理过程无关，所以在初始阶段可以采用较大的时间步长，以便计算能够快速收敛于周期解。一旦获得了周期解，则需要减小时间步长，以便能够获得满足时间精度要求的周期解。在采用二阶精度格式进行计算时，上述时间步长的变化幅度不应大于 20%，否则将会影响到解的精度。

10.5.4 滑动网格模型的后处理

滑动网格模型的后处理与其他非定常问题的后处理过程相同，首先需要读入某段时间的计算结果（算例文件和数据文件），然后再进行相关的后处理操作。对于空间周期性问题，虽然在计算过程中只需要计算一个子域的流场，但是在后处理过程中却可以根据需要，在 View（显示）面板中的 Periodic Repeats（周期性重复显示次数）下面，设置重复显示的次数，以便看到整个流场的计算结果。根据周期性假设，整个流场可以看作子域在空间上的重复，所以用这种方法可以检查计算是否合理，是否在空间上呈现出周期性流动特征。如果计算有误，那么在相邻子域的界面上会出现参数不连续的情况。这时应该反思一下，看看是否计算出现了错误。图 10-16 是一个转子定子流场，其周期性重复显示次数为 1，即子域的计算结果在显示窗口中重复显示了 2 次。从图中看，两个子域间参数分布是连续的，因此说明计算很好地满足了周期性要求。

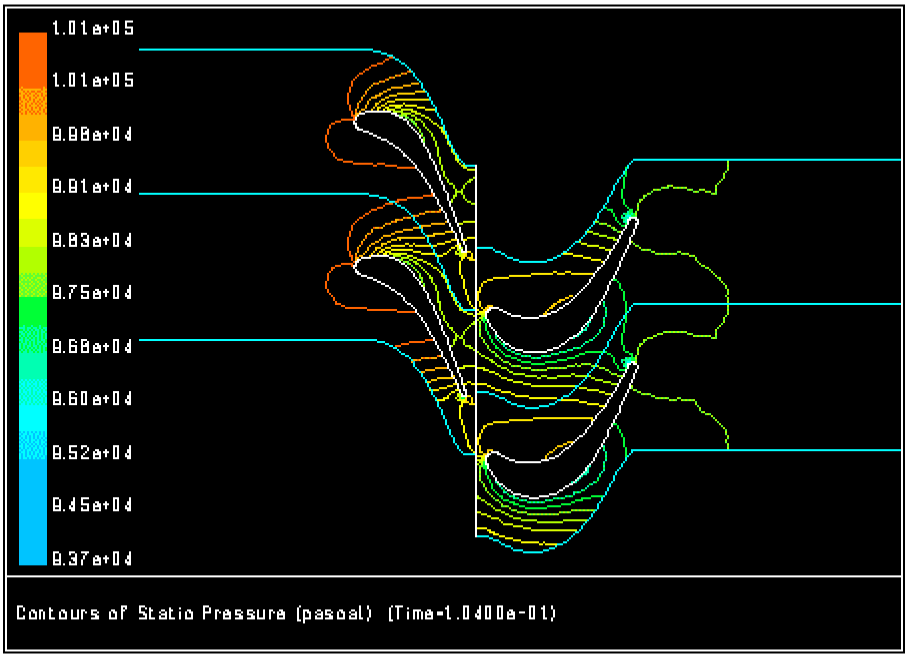


图 10-16 转子定子模型的静压等值线图

在显示速度矢量时，同样有绝对速度和相对速度两种形式。另外需要注意的是，后处理过程不能在交界区中的壁面、内部、周期等类型的边界上建立数据显示面（surface），但是可以在交界面上建立数据显示面，但结果将是单边的，就是只显示交界面一侧的结果。而且在跨越交接面时，等值线中可能会有细微的不连续。在画三维填充等值线时，图形中可能会出现一些小缝，但是这些缝只是图形显示问题，与解的连续性无关。

10.6 动网格模型

10.6.1 简介

动网格模型可以用来模拟流场形状由于边界运动而随时间改变的问题。边界的运动形式可以是预先定义的运动，即可以在计算前指定其速度或角速度；也可以是预先未做定义的运动，即边界的运动要由前一步的计算结果决定。网格的更新过程由 FLUENT 根据每个迭代步中边界的变化情况自动完成。在使用移动网格模型时，必须首先定义初始网格、边界运动的方式并指定参予运动的区域。可以用边界型函数或者 UDF 定义边界的运动方式。

FLUENT 要求将运动的描述定义在网格面或网格区域上。如果流场中包含运动与不运动两种区域，则需要将它们组合在初始网格中以对它们进行识别。那些由于周围区域运动而发生变形的区域必须被组合到各自的初始网格区域中。不同区域之间的网格不必是正则

的,可以在模型设置中用 FLUENT 软件提供的非正则或者滑动界面功能将各区域连接起来。

10.6.2 动网格守恒方程

在任意一个控制体中，广义标量 Φ 的积分守恒方程为：

$$\frac{d}{dt}\int_V \rho\Phi dV + \int_{\partial V}\rho\Phi\left(\vec{u}-\vec{u}_g\right)\cdot d\vec{A} = \int_{\partial V}\Gamma\nabla\Phi\cdot d\vec{A} + \int_V S_\Phi dV \qquad (10\text{-}7)$$

式中 ρ 为流体密度，$\vec{u}$ 为速度向量，$\vec{u}_g$ 移动网格的网格速度，Γ 为扩散系数，S_Φ 为源项，∂V 代表控制体 V 的边界。

方程（10-7）中的时间导数项，可以用一阶后向差分格式写成：

$$\frac{d}{dt}\int_V \rho\Phi dV = \frac{\left(\rho\Phi V\right)^{n+1} - \left(\rho\Phi V\right)^n}{\Delta t} \qquad (10\text{-}8)$$

式中 n 和 $n+1$ 代表不同的时间层。n+1 层上的 V^{n+1} 由下式计算：

$$V^{n+1} = V^n + \frac{dV}{dt}\Delta t \qquad (10\text{-}9)$$

式中 dV/dt 是控制体的时间导数。为了满足网格守恒定律，控制体的时间导数由下式计算：

$$\frac{dV}{dt} = \int_{\partial V}\vec{u}_g\cdot d\vec{A} = \sum_j^{n_f}\vec{u}_{g,j}\cdot\vec{A}_j \qquad (10\text{-}10)$$

式中 n_f 是控制体积的面网格数，$\vec{A}_j$ 为面 j 的面积向量。点乘 $\vec{u}_{g,j}\cdot\vec{A}_j$ 由下式计算：

$$\vec{u}_{g,j}\cdot\vec{A}_j = \frac{\delta V_j}{\Delta t} \qquad (10\text{-}11)$$

式中 δV_j 为控制体积面 j 在时间间隔 Δt 中扫过的空间体积。

10.6.3 动网格更新方法

动网格计算中网格的动态变化过程可以用三种模型进行计算，即弹簧光滑模型、动态层模型和局部重划模型。

1. 弹簧光滑模型

在弹簧光滑模型中，网格的边被理想化为节点间相互连接的弹簧。移动前的网格间距相当于边界移动前由弹簧组成的系统处于平衡状态。在网格边界节点发生位移后，会产生与位移成比例的力，力量的大小根据胡克定律计算。边界节点位移形成的力虽然破坏了弹簧系统原有的平衡，但是在外力作用下，弹簧系统经过调整将达到新的平衡，也就是说由弹簧连接在一起的节点，将在新的位置上重新获得力的平衡。从网格划分的角度说，从边界节点的位移出发，采用虎克定律，经过迭代计算，最终可以得到使各节点上的合力等于零的、新的网格节点位置，这就是弹簧光顺法的核心思想。图 10-17 和图 10-18 是弹簧光顺法的一个实例，图 10-17 是网格的初始状态，图 10-18 是网格在边界运动后达到新的平衡位置。

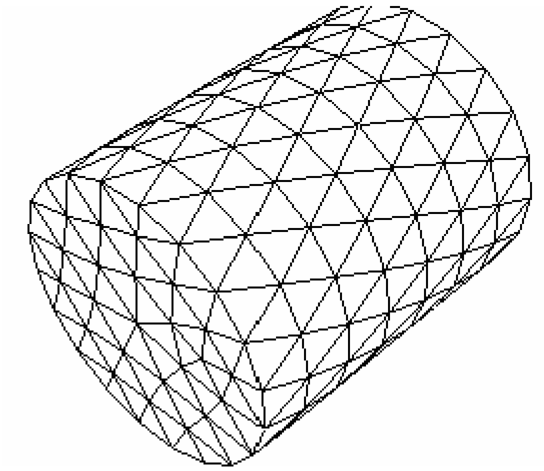


图 10-17 基于弹簧光滑节点开始状况

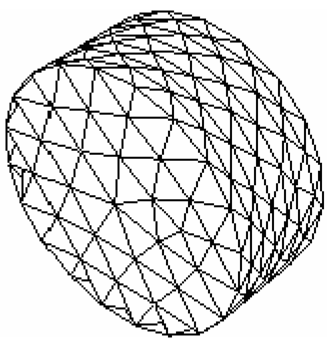


图 10-18 基于弹簧光滑节点结束状况

原则上弹簧光顺模型可以用于任何一种网格体系，但是在非四面体网格区域（二维非三角形），最好在满足下列条件时使用弹簧光顺方法：

（1）移动为单方向。

（2）移动方向垂直于边界。

如果两个条件不满足，可能使网格畸变率增大。另外，在系统缺省设置中，只有四面体网格（三维）和三角形网格（二维）可以使用弹簧光顺法，如果想在其他网格类型中激活该模型，需要在 dynamic-mesh-menu 下使用文字命令 spring-on-all-shapes?，然后激活该选项即可。

2. 动态层模型

对于棱柱型网格区域（六面体和或者楔形），可以应用动态层模型。动态层模型的中心思想是根据紧邻运动边界网格层高度的变化，添加或者减少动态层，即在边界发生运动时，如果紧邻边界的网格层高度增大到一定程度，就将其划分为两个网格层；如果网格层高度降低到一定程度，就将紧邻边界的两个网格层合并为一个层，如图 10-19 所示：

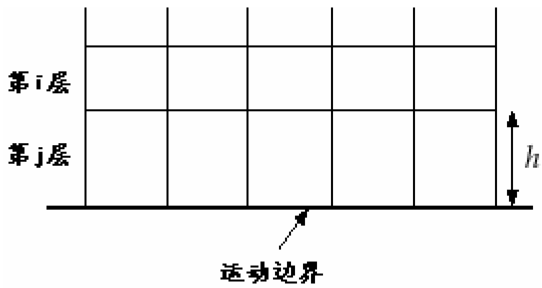


图 10-19 动态层

如果网格层 j 扩大，单元高度的变化有一临界值：

$$h_{\min} > (1+\alpha_s)h_0 \qquad (10\text{-}21)$$

式中 h_{min} 为单元的最小高度，h_0 为理想单元高度，α_s 为层的分割因子。在满足上述条件的情况下，就可以对网格单元进行分割，分割网格层可以用常值高度法或常值比例法。

在使用常值高度法时，单元分割的结果是产生相同高度的网格。在采用常值比例法时，网格单元分割的结果是产生是比例为 α_s 的网格。图 10-20 与 10-21 是用常值高度法和常值比例法的结果。

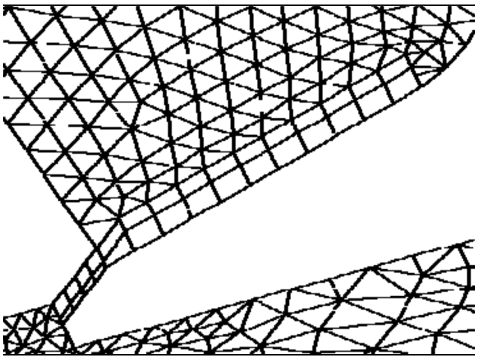


图 10-20 用常值高度法分割网格

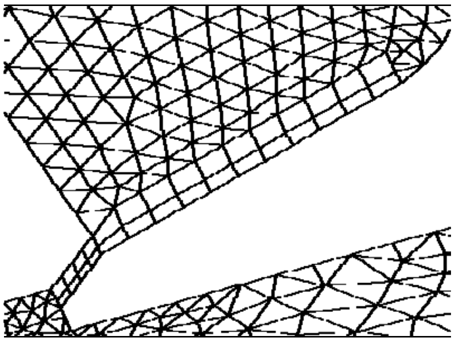


图 10-21 用常值比例法分割网格

若对第 j 层进行压缩，压缩极限为：

$$h_{\min} < \alpha_c h_0 \quad (10\text{-}22)$$

式中 α_c 为合并因子。在紧邻动边界的网格层高度满足这个条件时，则将这一层网格与外面一层网格相合并。

动网格模型的应用有如下限制：

（1）与运动边界相邻的网格必须为楔形或者六面体（二维四边形）网格。

（2）在滑动网格交界面以外的区域，网格必须被单面网格区域包围。

（3）如果网格周围区域中有双侧壁面区域，则必须首先将壁面和阴影区分割开，再用滑动交界面将二者耦合起来。

（4）如果动态网格附近包含周期性区域，则只能用 FLUENT 的串行版求解，但是如

果周期性区域被设置为周期性非正则交界面，则可以用 FLUENT 的并行版求解。

如果移动边界为内部边界，则边界两侧的网格都将作为动态层参予计算。如果在壁面上只有一部分是运动边界，其他部分保持静止，则只需在运动边界上应用动网格技术，但是动网格区与静止网格区之间应该用滑动网格交界面进行连接，如图 10-22 所示。

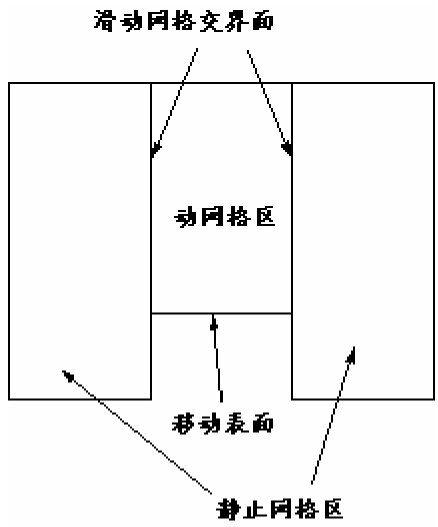


图 10-22 动态层区域与临近区域间应用滑动界面

局部重划模型

在使用非结构网格的区域上一般采用弹簧光顺模型进行动网格划分，但是如果运动边界的位移远远大于网格尺寸，则采用弹簧光顺模型可能导致网格质量下降，甚至出现体积为负值的网格，或因网格畸变过大导致计算不收敛。为了解决这个问题，FLUENT 在计算过程中将畸变率过大，或尺寸变化过于剧烈的网格集中在一起进行局部网格的重新划分，如果重新划分后的网格可以满足畸变率要求和尺寸要求，则用新的网格代替原来的网格，如果新的网格仍然无法满足要求，则放弃重新划分的结果。

在重新划分局部网格之前，首先要将需要重新划分的网格识别出来。FLUENT 中识别不合乎要求网格的判据有二个，一个是网格畸变率，一个是网格尺寸，其中网格尺寸又分最大尺寸和最小尺寸。在计算过程中，如果一个网格的尺寸大于最大尺寸，或者小于最小尺寸，或者网格畸变率大于系统畸变率标准，则这个网格就被标志为需要重新划分的网格。在遍历所有动网格之后，再开始重新划分的过程。局部重划模型不仅可以调整体网格，也可以调整动边界上的表面网格。

需要注意的是，局部重划模型仅能用于四面体网格和三角形网格。在定义了动边界面以后，如果在动边界面附近同时定义了局部重划模型，则动边界上的表面网格必须满足下列条件：

（1）需要进行局部调整的表面网格是三角形（三维）或直线（二维）。

（2）将被重新划分的面网格单元必须紧邻动网格节点。

（3）表面网格单元必须处于同一个面上并构成一个循环。

（4）被调整单元不能是对称面（线）或正则周期性边界的一部分。

动网格的实现在 FLUENT 中是由系统自动完成的。如果在计算中设置了动边界，则 FLUENT 会根据动边界附近的网格类型，自动选择动网格计算模型。如果动边界附近采用的是四面体网格（三维）或三角形网格（二维），则 FLUENT 会自动选择弹簧光顺模型和局部重划模型对网格进行调整。如果是棱柱型网格，则会自动选择动态层模型进行网格调整。在静止网格区域则不进行网格调整。

动网格问题中对于固体运动的描述，是以固体相对于重心的线速度和角速度为基本参数加以定义的。既可以用型函数定义固体的线速度和角速度，也可以用 UDF 来定义这两个参数。同时需要定义的是固体在初始时刻的位置。

10.6.4 动网格问题的建立

设置动网格问题的步骤如下：

（1）在 Solver（求解器）面板中选择非定常流（unsteady）计算。

（2）设定边界条件，即设定壁面运动速度。

（3）激活动网格模型，并设定相应参数，菜单操作如下：

Define -> Dynamic Mesh -> Parameters...

（4）指定移动网格区域的运动参数，菜单操作如下：

Define -> Dynamic Mesh -> Zones...

（5）保存算例文件和数据文件。

（6）预览动网格设置，菜单操作为：

Solve -> Mesh Motion...

（7）在计算活塞问题时，设定活塞计算中的事件：

Define -> Dynamic Mesh -> Events...

并可以通过显示阀与活塞的运动，检查上述设置是否正确：

Display -> IC Zone Motion...

（8）应用自动保存功能保存计算结果。

File -> Write -> Autosave...

在动网格计算中，因为每个计算步中网格信息都会改变，而网格信息是储存在算例文件中的，所以必须同时保存算例文件和数据文件。

（9）如果想建立网格运动的动画过程，可以在 Solution Animation（计算结果动画）面板中进行相关设置。

10.6.6 设定动网格参数

为了使用动网格模型，需要在 dynamic mesh（动网格）面板中激活 Dynamic Mesh（动网格）选项。如果计算的是活塞运动，则同时激活 In-Cylinder（活塞）选项。然后选择动网格模型，并设置相关参数。如果激活了活塞运动，则同时设置活塞运动的相关参数，如图 10-23 所示。

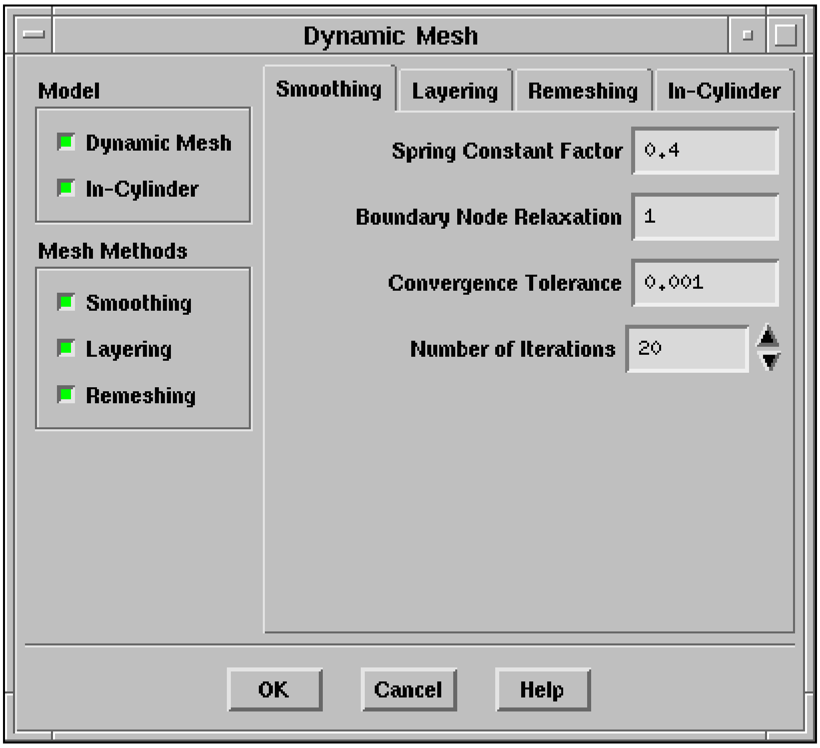


图 10-23 动网格面板

1. 选择网格更新模型

在 Mesh Methods（网格划分方法）下面选择 Smothing（弹簧光顺模型），Layering（动态层模型）和（或）Remshing（局部重划模型）。

2. 设置弹簧光顺参数

激活弹簧光顺模型，相关参数设置位于 Smoothing（光顺）标签下，可以设置的参数包括 Spring Constant Factor（弹簧弹性系数）、Boundary Node Relaxation（边界点松弛因子）、Convergence Tolerance（收敛判据）和 Number of Iterations（迭代次数）。

弹簧弹性系数的影响如图 10-24 与图 10-25 所示：弹性系数应该在 0 到 1 之间变化，

弹性系数等于 0 时，弹簧系统没有耗散过程，在图中算例中，靠近壁面的网格没有被改变，而是保持了原来的网格形状和密度；在弹性系数等于 1 时，弹簧系统的耗散过程与缺省设置相同，从图中可以发现壁面发生变形，壁面附近网格因为过度加密而质量下降。因此在实际计算中应该在 0 到 1 之间选择一个适当的值。

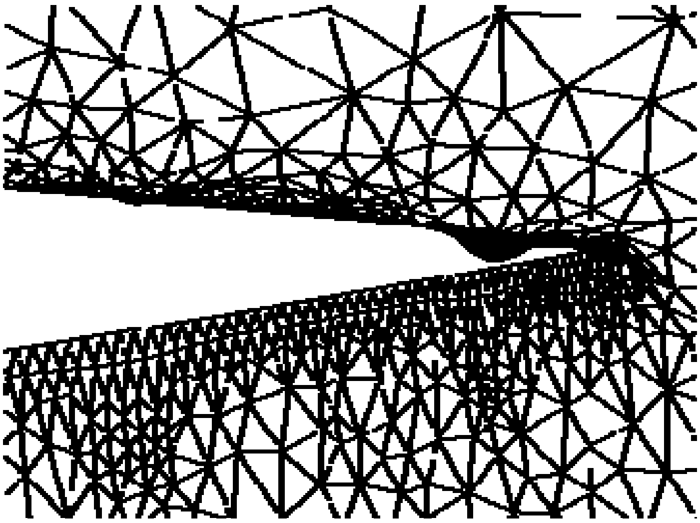


图 10-24 弹性系数为 1 时的网格

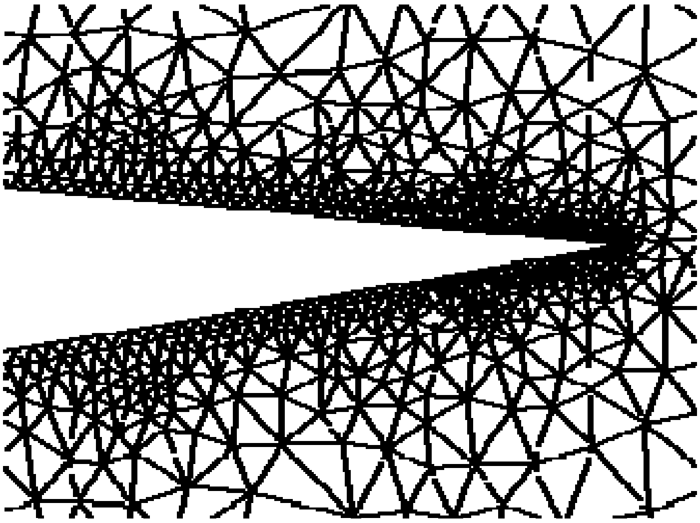


图 10-25 弹性系数为 0 时的网格

边界点松弛因子用于控制动边界上网格点的移动。当这个值为零时，边界节点不发生移动；在这个值为 1 时，则边界节点的移动计算中不采用松弛格式。在大多数情况下，这个值应该取为 0 到 1 之间的一个值，以保证边界节点以合适的移动量发生移动。

收敛判据就是网格节点移动计算中，迭代计算的判据。迭代次数是指网格节点移动计

算的最大迭代次数。

3. 动态层

在 Layering（动态层）标签下，可以设置与动态层模型相关的参数。通过设定 Constant Height（常值高度）与 Constant Ratio（常值比例）可以确定分解网格的两种方法。Split Factor（分割因子）和 Collapse Factor（合并因子）则分别为式（10-21）中的α_s和式（10-22）中的α_c。

4. 局部重新划分网格

在 Remeshing（重划网格）标签下，设置与局部重划模型相关的参数。可以设置的参数包括 Maximum Cell Skewness（最大畸变率）、Maximum Cell Volume（最大网格体积）和 Minimum Cell Volume（最大网格体积），其含义如前所述，主要用于确定哪些网格需要被重新划分。在缺省设置中，如果重新划分的网格优于原网格，则用新网格代替旧网格；否则，将保持原网格划分不变。如果无论如何都要采用新网格的话，则可以在 Options（选项）下面选择 Must Improve Skewness（必须改变畸变率）选项。

如果 Options（选项）下面的 Size Function（尺寸函数）被激活，则还可以用网格尺寸分布函数标志需要重新划分的网格。假设在某点附近的理想网格尺寸为L，而某个网格的尺寸为L'，如果：

$$L' \notin \left(\frac{4}{5}\gamma L, \frac{5}{4}\gamma L\right) \tag{10-23}$$

则网格被标志为需要重新划分的网格，并在随后的计算中被重新划分。式（10-23）中的γ用下面公式计算：

当$\alpha > 0$时，$\gamma = 1 + \alpha d_b^{1+2\beta}$ （10-24）

当$\alpha < 0$时，$\gamma = 1 + \alpha d_b^{\frac{1}{1-\beta}}$ （10-25）

式中d_b为网格到壁面边界的最小距离，α和β就是需要设置的 Size Function Variation（尺寸函数增量）和 Size Function Rate（尺寸函数变化率）。

在（10-24）和（10-25）两个公式中，α以边界网格尺寸为基准控制内部网格的大小，α等于 0.5 表示内部网格的尺寸至少是边界网格的 1.5 倍，α等于-0.5 表示内部网格的尺寸等于边界网格的 0.5 倍，如果α等于 0 则表示内部网格与边界网格大小相同。β的取值在－1 和 1 之间，用于控制网格从边界到内部区域的变化速率。β取正值表示网格变化速率

较慢，取负值则表示变化速率较快，取 0 则表示网格从边界到内部区域呈线性变化。

另外一个选项 Size Function Resolution（尺寸函数分辨率）用于定义尺寸分布函数对网格大小的分辨率，其参照值是最小网格尺寸。

点击 Use defaults（使用缺省值）按钮，可以恢复系统缺省设置。

5. 设定活塞运动参数

如果在计算中选择使用 In-cylinder（活塞）模型，需要指定 Crank Shaft Speed（曲柄速度）、Starting Crank Speed（曲柄起始速度）、Crank Period（曲柄周期）以及 Crank Angle Step Size（曲柄角度时间步长）。

FLUENT 中还提供一个内建函数用于计算活塞位置。这个函数的自变量为曲柄转角，如果选用这个函数，还需要指定 Piston Stroke（活塞行程）和 Connecting Rod Length（连接杆长度）二个参数。

10.6.7 定义动网格的运动方式

在计算动网格问题时，必须定义动网格区的运动方式。在动网格区为刚体运动时，可以用型函数和 UDF 来定义其运动；在动网格区为变形区域时，则需要定义其几何特征及局部网格重划参数；如果动网格区既做刚体运动又有变形发生，则只能用 UDF 来定义其几何形状的变化和运动过程。

上述定义在 Dynamic Zones（动态区域）面板中设置，启动该面板的菜单操作顺序为：

Define -> Dynamic Mesh -> Zones...

在这个面板中可以修改动态区域的设置、计算刚体运动区域的重心或删除一个动态区域。方法是首先在 Dynamic Zones（动态区域）列表中选择一个动网格区，然后修改其设置参数，或计算其重心，或进行删除操作，最后点击 Create（创建）按钮保存设置。

对于新加入的区域，需要先从 Zone Names（区域名称）下选择相关区域，然后在 Type（类型）下选择其运动类型。可供选择的运动类型包括 Stationary（静止）、Rigid Body（刚体运动）、Deforming（变形）和 User-Defined（用户自定义）四种。

1. 静止区域设置

如果被指定区域为静止区域，则首先在 Zone Names（区域名称）下选择这个区域，然后在 Type（类型）下选择 Stationary（静止），再指定 Adjacent Zone（相邻区域）的 Cell Height（网格高度）用于网格重新划分，最后点击 Create（创建）按钮完成设置。

2. 刚体运动区域设置

如果被指定区域为刚体运动区域，则其设置过程如下：

（1）在 Zone Names（区域名称）下选择这个区域的名称，然后在 Type（类型）下选

择 Rigid Body（刚体）。

（2）在 Motion Attributes（运动属性）标签下的 Motion UDF/Profile（用 UDF 或型函数定义运动）中确定究竟用型函数，还是 UDF 来做运动定义。

（3）在 C.G. Location（重心位置）中定义刚体重心的初始位置。

（4）在 C.G. Orentation（重心方向）中定义重力在惯性系中的方向。

（5）如果计算中包含活塞计算，则需要在 Valve/Piston Axis（阀门或活塞轴）中指定阀门或活塞的参考轴。

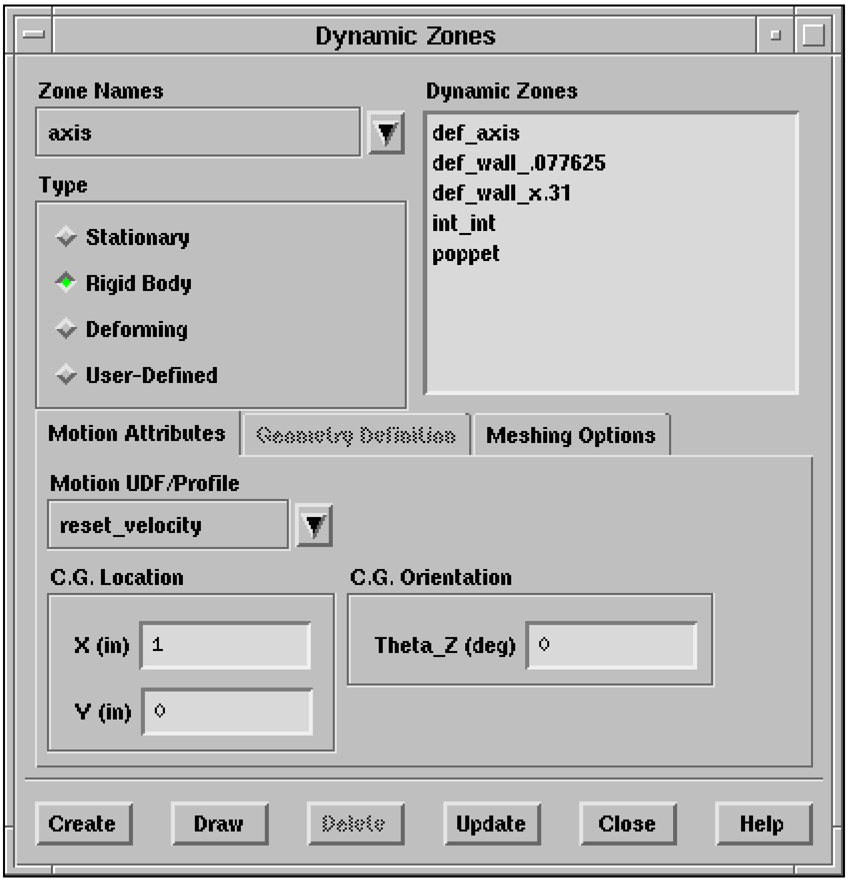


图 10-26 Dynamic Zones（动态区域）面板

如果在所定义的网格区域中，某种形状的网格单元需要被排除在上述设置之外，则可以在 Motion Mask（运动屏蔽）中选择这些单元形状。

（6）如果所定义的区域是面区域，则还需要定义 Cell Height（网格高度）。这个参数用于定义局部网格重划时，与面区域相邻的网格的理想高度。

（7）点击 Create（创建）按钮，完成设置。

3. 变形运动

变形区域的设置过程为：

（1）在 Zone Names（区域名称）中选择区域，并在 Type（类型）下选择 Deforming（变形）。

（2）在 Geometry Definition（几何定义）下定义变形区的几何特征，共有 4 个选项：如果没有合适的几何形状，就在 Definition 中选择 none；如果变形区为平面，则选择 plane（平面）并在 Point on Plane（平面上的点）中定义平面上一点，同时在 Plane Normal（平面法向）中定义法线方向；如果变形区为圆柱面，则选择 Cylinder（圆柱），并同时定义 Cylinder Radius（圆柱半径）、Cylinder Origin（圆柱原点）和 Cylinder Axis（圆柱轴）；如果变形区几何形状需要用 UDF 来定义，则在 Definition（定义）中选择 user-defined（用户定义），并在 Geometry UDF（几何 UDF）中选择适当的函数。

（3）在 Remeshing Options（重划选项）标签下定义与网格局部重划相关的参数。重划方法在 Mesh Methods（网格划分方法）中选择，其中包括 Smoothing（弹簧光顺模型）、Layering（动态层模型）和 Remeshing（网格重划模型）。如果动网格区域为面域，则需要设置局部重划模型中的几个参数，包括 Height（网格高度）、Height Factor（高度因子）和 Maximum Skewness（最大畸变率）。如果动网格区域为体积域时，还可以设置 Minimum Volume（最小体积）、Maximum Volume（最大体积）和 Maximum Skewness（最大畸变率），这项设置可以覆盖系统缺省设置，从而改变网格重划过程。

（4）点击 Create（创建）按钮完成设置。

4. 用户定义的运动方式

对于同时存在运动和变形的区域，只能使用 UDF 来定义其运动方式，定义步骤如下：

（1）在 Zone Names（区域名称）中选择需要定义的区域名称，并在 Type（类型）下选择 User-Defined（用户定义）。

（2）在 Motion Attributes（运动属性）标签下，然后在 Mesh Motion UDF（网格运动 UDF）下选择相应的 UDF 函数。

（3）点击 Create（创建）按钮完成设置。

关于 UDF 函数的编制方法请参见与 UDF 相关的章节。

10.6.8 预览动网格

在设置好动网格模型及动网格区的运动方式后，可以通过预览的方式检查设置效果。预览功能在 Mesh Motion（网格运动）面板中进行设置，启动这个面板的菜单操作为：

Solve -> Mesh Motion...

预览操作步骤如下：

（1）在参数设置完毕后，首先保存算例（case）文件。因为与网格设置有关的记录都

保存在算例文件中，在预览过程中伴随着网格的更新，与网格有关的记录也不断被刷新，如果不进行保存，则无法恢复原始设置状态，在发现参数设置问题后就无法进行更改了。

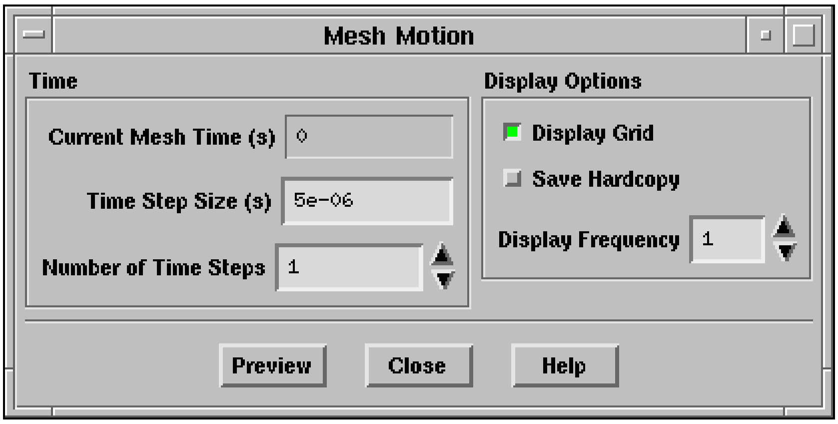


图 10-27 Mesh Motion（网格运动）面板

（2）设置迭代时间步数和时间步长。在计算过程中，当前时间将被显示在 Current Mesh Time（当前网格时间）栏中。如果在计算中选用了活塞模型选项，则时间步长用曲柄速度（shaft speed）和曲柄转动步长（crank angel step size）计算得出。

（3）为了在图形窗口中预览网格变化过程，需要激活 Display Options（显示选项）下的 Display Grid（显示网格），并在 Display Frequency（显示频率）中设置显示频率，即每分钟显示图幅数量。如果要保存显示的图形，则同时激活 Save Hardcopy（保存硬拷贝）选项。

（4）点击 Preview（预览）按钮开始预览。

在定义了活塞运动时，活塞运动的预览是在 IC Zone Motion（活塞运动）面板中实现的，激活这个面板的菜单操作为：

Display -> IC Zone Motion...



图 10-28 IC Zone Motion（活塞运动）面板

预览操作的具体步骤为：

（1）在 Display Grid（显示网格）面板中选择准备预览的网格区域。

（2）在 IC Zone Motion（网格运动）面板中，设置曲柄角度增量（Increment）和迭代步数（Number of Steps）。

（3）点击 Preview（预览）按钮开始预览。

10.6.9 定义活塞事件

在计算活塞运动时，可以通过定义活塞事件（events）的方式，在计算中增加问题的复杂度，使计算更接近真实情况。这些事件都是以曲柄角度为自变量的，比如可以将打开排气阀的时间定义为曲柄角度的函数，然后在曲柄转到这个角度时，开始打开排气阀操作，等等。下面介绍定义活塞事件的具体步骤和事件种类，从这些介绍中可以了解活塞事件的基本概念和使用方法。

1. 活塞事件的设置步骤

活塞事件在 Dynamic Mesh Events（动网格事件）面板中定义，如图 10-29 所示。

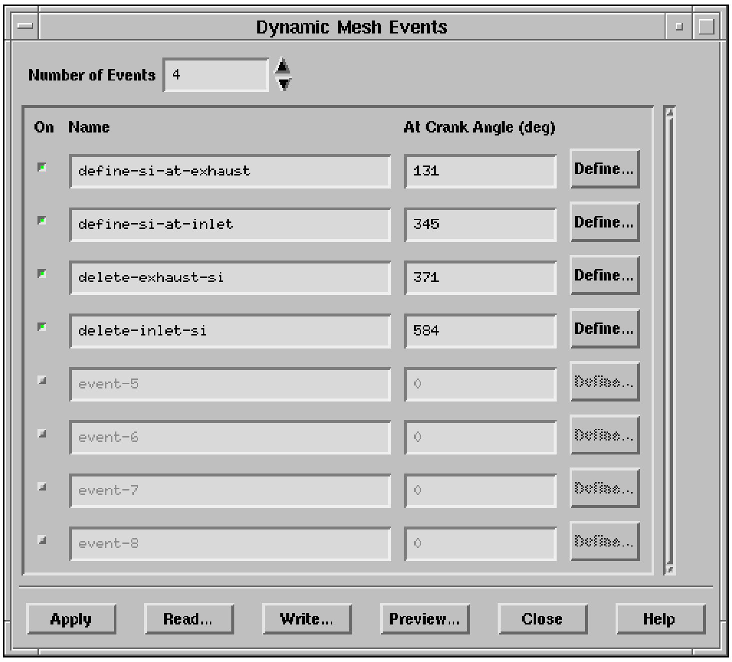


图 10-29 Dynamic Mesh Events（动网格事件）面板

启动这个面板的菜单操作次序为：

Define -> Dynamic Mesh -> Events...

活塞事件设置步骤为：

（1）在 Number of Events（事件数量）栏中输入需要定义的事件数量。

（2）在 Name（名称）下面输入事件名称。

（3）在 At Crank Angel（曲柄角度）下输入事件发生的曲柄角度。

（4）点击 Define（定义）按钮进入 Define Event（定义事件）面板做详细定义，如图 10-30。

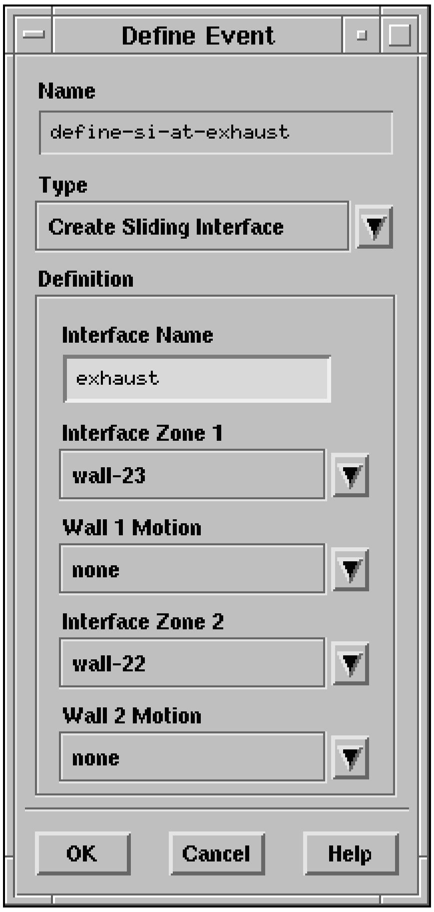


图 10-30 Define Event（定义事件）面板

（5）在 Type（类型）下面选择动网格运动类型，可供选择的运动类型包括 Change Zone Type（改变网格区域类型）、Copy Zone BC（拷贝网格区域边界条件）、Create Sliding Interface（创建滑动网格交界面）、Delete Sliding Interface（删除滑动网格交界面）、Change Motion Attribute（改变运动属性）、Change Time Step Size（改变时间步长）、Insert Boundary Layer（插入边界层）、Remove Boundary Layer（删除边界层）、Insert Interior Layer（插入内部网格层）和 Remove Interior Layer（删除内部网格层）等。在确定了事件类型后，还需要对所选运动类型做进一步设置，详细情况请参见下一小节中的内容。

（6）对其他事件重复（2）到（5）的设置过程。

（7）将所有事件设置完毕后，点击 Apply（应用）按钮保存所有设置。

（8）在 Dynamic Mesh Events（动网格事件）面板中点击 Preview（预览）按钮，预览上述设置的事件。

2. 活塞事件

在上一小节中讲到的各种事件类型的含义如下：

（1）Change Zone Type（改变网格区域类型）。在计算过程中可以改变网格区域的类型，即将网格类型在 wall（壁面）、interface（交界面）、fluid（流体）和 solid（固体）等类型之间进行转换。在 New Zone Type（新网格区域类型）中指定一种类型，则当曲柄运动到相应位置时，指定网格区域类型就会发生转换。

（2）Copy Zone BC（拷贝网格区域边界条件）。在网格区域类型改变后，原来设置在网格上的边界条件也要发生相应的变化，这种变化可以通过拷贝另一个区域的边界条件实现，即在设定改变网格区类型的同时，设定拷贝边界条件事件，在网格类型发生改变时，随即从另一个边界上拷贝其边界条件作为新类型下网格区域的边界条件。

（3）Creating Sliding Interface（创建滑动网格交界面）。在选择了这个事件类型后，需要定义构成交界面的两个面，即两个相邻网格区域 Interface Zone 1（交界面区 1）和 Interface Zone 2（交界面区 2）。在这里不用考虑界面类型转换的问题，系统会自动将定义为交界面区的网格区域转换为内部面类型。

（4）Deleting Sliding Interface（删除滑动网格交界面）。在选择了这个事件类型后，需要在 Interface Name（界面名称）中指定将被删除的交界面。在删除界面后，界面所在网格区域将被自动转换为壁面，如果不准备采用系统缺省设置的壁面条件的话，可以考虑如（2）中提示的那样，从其他壁面上拷贝边界。

（5）Change Motion Attribute（改变运动属性）。可以用这个事件类型修改网格运动类型。选择了这个事件类型后，需要在 Attribute（运动属性）中指定一种运动类型，可以选择的类型包括 sliding（滑动）、moving（移动）和 remesh（网格重划），同时在 Status（状态）中选择 enable（激活）或 disable（关闭）。

（6）Change Time Step（改变时间步长）。在 Crank Angel Step Size（曲柄角度步长）中输入新的步长，可以修改迭代的时间步长）。

（7）Insert Boundary Cell Layer（插入边界网格层）。选择这个事件类型后，需要指定 Base Dynamic Zone（基础动网格区）和 Side Dynamic Zone（临近动网格区）。边界层网格将以基础动网格区为起点生成，临近动网格区就是网格即将发生变形的区域。

（8）Deleting Boundary Cell Layer（删除边界网格层）。这个事件类型选项与（7）中的事件相对应，在选择这个选项后，只要指定 Base Dynamic Zone（基础动网格区）就可以将插入的边界网格删除。

（9）Insert Interior Cell Layer（插入内部网格层）。如同插入边界网格层一样，这项操作也需要指定 Base Dynamic Zone（基础动网格区）和 Side Dynamic Zone（临近动网格区），

不同的是还需要为新生成的界面命名，即指定 Internal Zone 1 Name（内部网格区 1 名称）和 Internal Zone 2 Name（内部网格区 2 名称）。

（10）Remove Interior Cell Layer（删除内部网格层）。选择这个事件选项后，只要指定被删除的网格区名称，Internal Zone 1 Name（内部网格区 1 名称）和 Internal Zone 2 Name（内部网格区 2 名称）即可。

3. 输入输出活塞事件

在 Dynamic Mesh Events（动网格事件）面板上，点击 Write...（输出）按钮可以将设置好的活塞事件以文件形式输出到磁盘上；点击 Read...（输入）按钮可以将活塞事件文件读入计算模型。

4. 活塞计算中初始网格的生成

可以在 dynamesh-mesh-control 文本菜单下，用文本命令 position-starting-mesh 设置活塞的初始位置，例如：

```
/define/models/dynamic-mesh-controls> position-starting-mesh
Start Crank Angle (deg) [0] 340
```

即将初始位置设置在曲柄转角为 340°处。

在进行活塞计算的设置中，需要妥善处理动网格的拓扑关系、活塞和阀门的运动属性、进气阀和排气阀的开关以及活塞事件的先后次序，尽量全面、合理地进行设置，以保证计算能顺利完成。

第 11 章 为数据显示、报告创建表面

在 Fluent 中可以方便的选择进行可视化流场的区域。这些区域称为表面，由很多方式可以创建表面。对于 3D 问题，因为不可能对整个区域进行矢量，等值线，XY 曲线的绘制，所以必须要创建表面来进行相关操作。另外一种情况是对于无论是 2D 还是 3D，如果希望创建表面积分报告，必须要建立表面。下面的章节将集中讲解如何创建，重命名，组合，删除和确定尺寸等操作。

11.1 应用表面

前面已经提到，应用表面主要有两种情况：

1. 要可视化 3D 问题的内部流场和创建 XY 曲线，必须要选择流场区域（表面）。
2. 建立表面积分报告。

所有的表面都储存于 Case 文件中，各种表面的功能如下：

1．区域表面：

如果要查看边界上的结果，可以创建这种区域表面，它的特点是和存在的单元/面区域包含相同的单元。

2．分割表面：

对于并行 Fluent，创建由不同网格间隔定义的分割表面有极其方便的应用。

3．点表面：

可以通过创建仅包含一个点的表面，来监视流场区域中一个特定点变量或函数的变化。

2．线面和耙状表面

要显示流动轨迹，必须要指定颗粒的释放表面。线面和耙状表面非常适合此用途，而且所得结果可以方便的和风洞结果进行比较。所谓的耙状表面是指在指定的两个端点之间包含等间距的指定数目的点。线表面是包含指定端点的并且延伸穿过整个区域的线，数据点位于该线穿过单元的中心因此并不是等间距的。

4．平面表面：

平面表面是通过三个指定点的平面。

5． 二次表面

可以通过指定二次函数的系数来定义，线(2D)，平面(3D)，圆(2D)，球(3D)，或者二次表面等。

6．常值表面：

常指表面的功能是显示对于指定变量有相同单元的表面。产生常数表面需要指定坐标轴。这里要注意，根据压力常数生成的常数表面也可以显示其他变量。

11.2 区域表面

所有的边界都会自动生成区域表面，所以除非不小心删去一个，不需要去生成一个这样的面。

创建区域表面的步骤如下：

1. 在 Zone Surface Panel 中

面板打开步骤为，Surface -〉Zone...

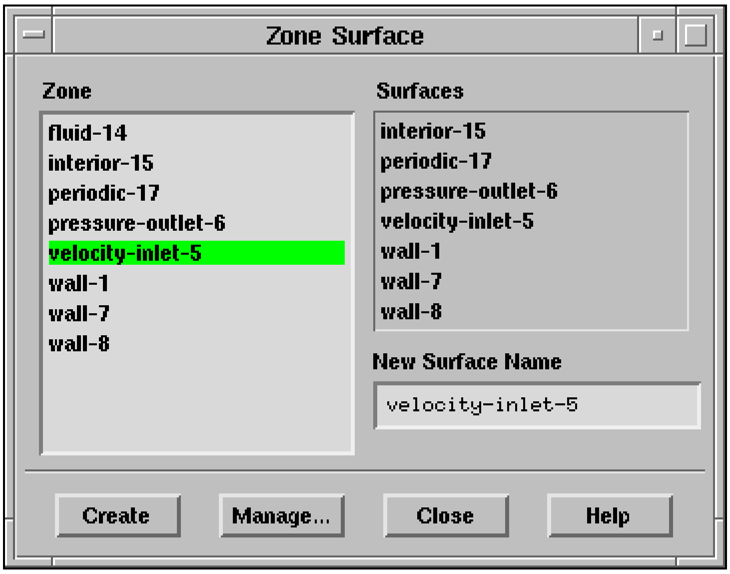


图 11-2-1 区域表面面板

（1） 在 zone 列表中选择要建立表面的区域。

（2） 选择输入新名字或者用默认值，

（3） 点击 creat 按钮，建立新的表面。

点击 manage 可以进行删除和其他操作。

11.3 分块表面

对于并行 Fluent 运算，分块表面尤其适用。因为用分块表面分开的网格，可以在不同的处理器上进行运算。分块表面包含交接边界上的两个网格单元。图 11.3.1 是为分割单元等值图，与网格图重叠。

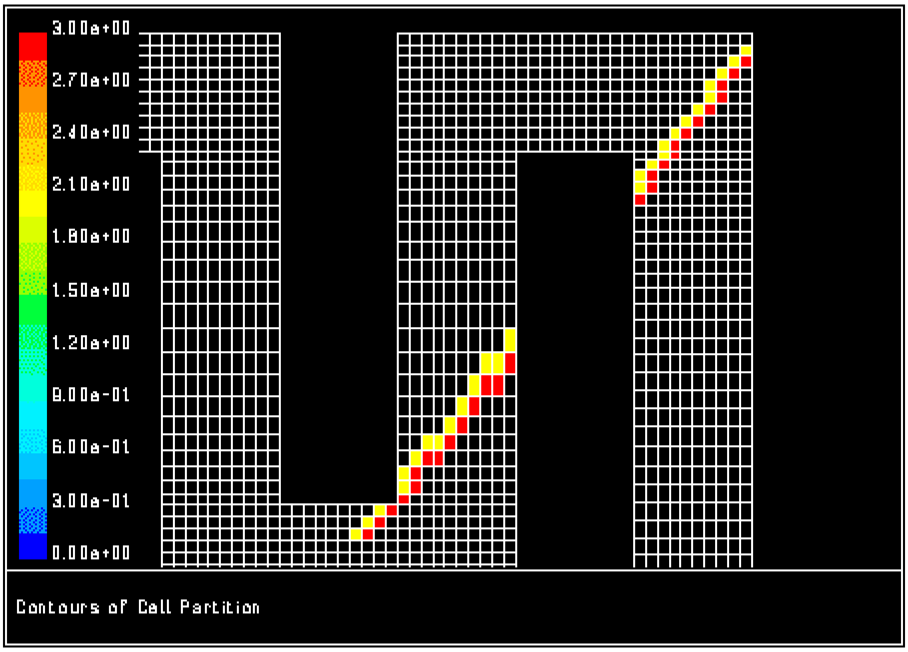


图 11-3-1 分割单元等值线图，与网格重叠

创建分割表面的步骤如下：

1. 打开分割表面面板，surface->partition 图 11.3.2 分割表面面板

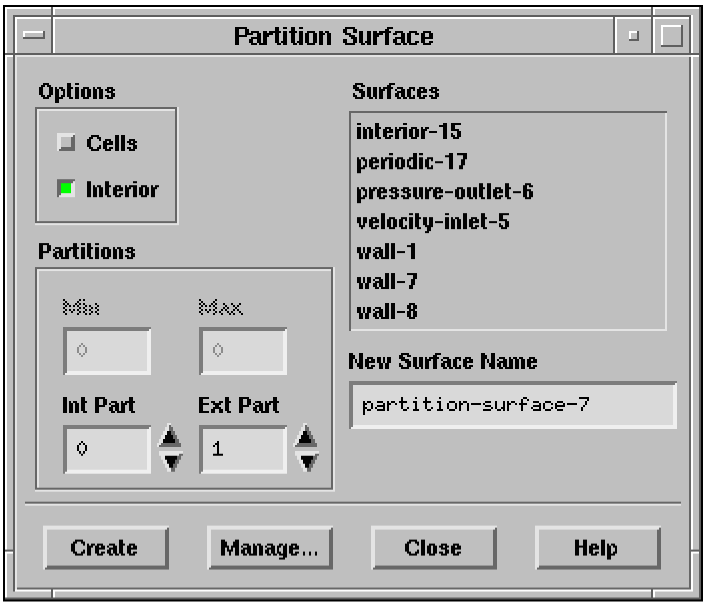


图 11-3-2 分割表面面板

2. 在 partitions 标题下通过指定两个边界部分而指定分割边界。定义分割表面的边界为“内部”与“外部”的边界。面板中的 IntPart 为内部的 ID 号，ExtPart 为外部的 ID 号。最小（Min）最大（Max）为最大的分区数目。如果包含超过两个部分，每一个部分都会和几个外部部分共享边界。通过适当的设定 Int Part 和 Ext part 可以为任意的边界创建表面。

3.　通过选和不选 options 下的 cells 和 interior 来决定分割表面的内部和外部面与单元。例如，要得到包含分割边界内部单元的表面，可以选择 cells 和 interior。如果要创建包含外部单元的边界，则选择 cells 并不选 interior。如果希望创建的表面包含边界上的面而不是单元，则应不选 cells。
4.　可以在 new surface name 中输入新的名字。
5.　点击创建按钮。新的表面名字会添加到 Surfaces 列表中。

11.4 点表面

如果要显示流域中一个点的数据，就必须要使用点表面。在显示节点数据时，数据点上显示的数据为该点相连结点的线性平均值。显示单元值数据时，会显示该点的数据。创建点表面的过程如下：

1. 打开点表面面板。

Surface->point

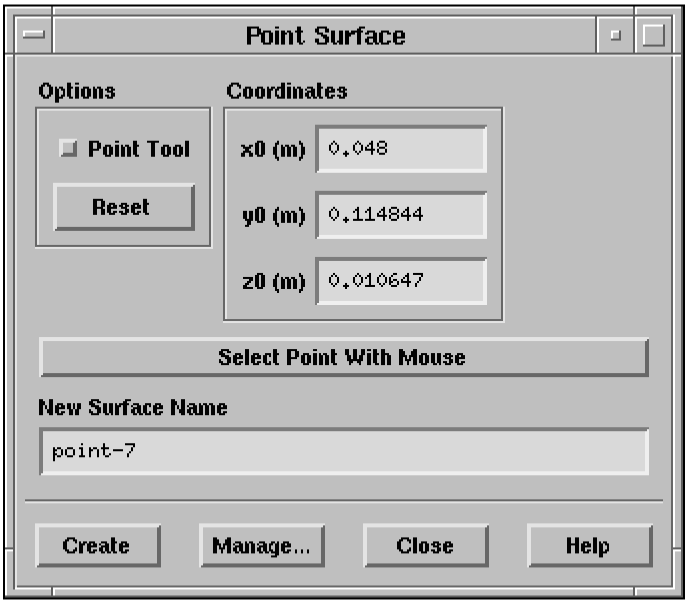


图 11-4-1 点表面面板

2. 指定点的位置，有三种方式：
（1）　输入点的坐标
（2）　用鼠标选定点。
（3）　用(点工具)point tool 选定点。
3.　可以输入新名字。
4.　点击创建按钮，完成点表面创建过程。

11.4.1 应用点工具

应用点工具可以方便的在图形窗框中交互创建准确创建点。从起始点出发，可以方便的移动该点到目的点。例如，如果需要管道中央的数据点，可以首先在邻近区域选择点，然后再把该点移动到目的位置。

1．初始化点工具

在应用点工具前，首先设定坐标到合适的初始位置。可以用输入坐标方法或者鼠标选择法。一旦选择指定点，则可以激活点工具。点工具，八边的多面体，会出现。如图 11.4.2：

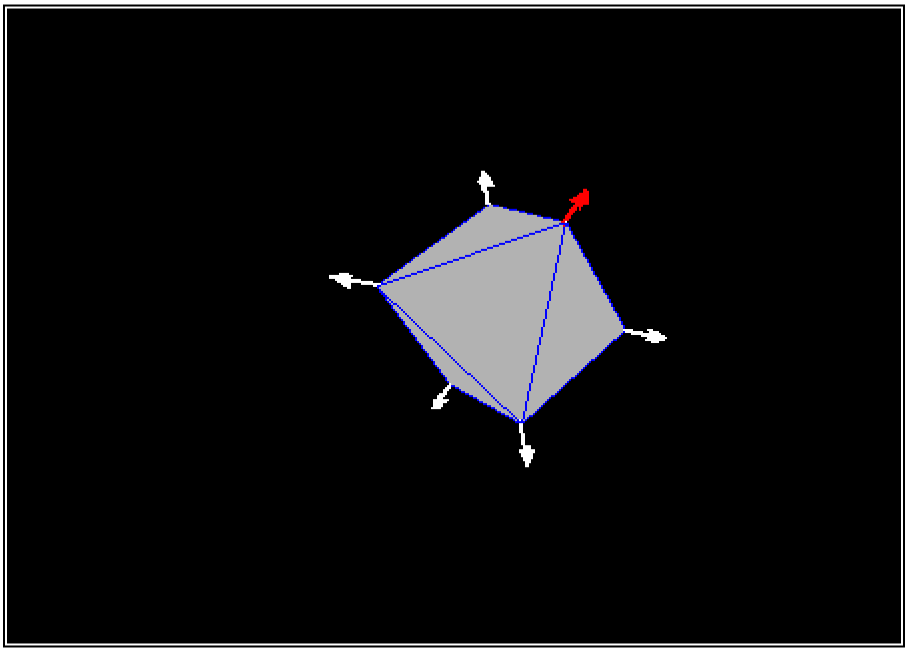


图 11-4-2 点工具

2．移动点工具

要沿着红轴方向移动点,需要首先点击鼠标的探测键,然后就可以拖拽该点到任何位置。要沿着不同的方向的方向拖动点，需要用<shift>键。

3．重新设定点工具

如果找不到点工具，可以按下 Reset 按钮，或者关闭点工具然后重新初始化。

11.5 线和耙表面

为了和风洞试验结果比较，可以用创建线和耙表面释放粒子来显示整个流动的情况。耙表面包含指定数目的平均分布的端点。而线表面则只包含指定的端点并延伸至整个流域。

创建线或者耙表面的步骤如下：

1． 打开 line/rake surface 面板。

Surface->line/rake

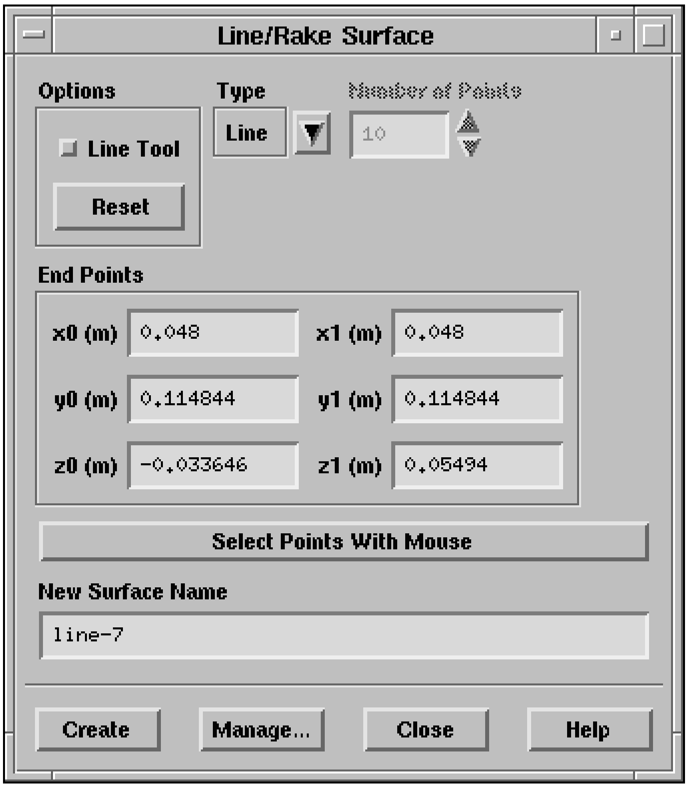


图 11-5-1 line/rake 表面面板

2. 在 type 下拉菜单中选择要创建的表面类型。
3. 若创建耙表面，指定等间距分布的点数。
4. 指定位置，有三种方式：
1） 输入第一点的坐标和第二点的坐标。
2） 用鼠标选择方式。
3） 用线工具互动的精确定位。
5. 可以选择输入新名字
6. 点击 creat 完成创建。

11.5.1 应用线工具

与应用点工具类似，应用线工具可以方便，互动的精确定位线的位置。从一条初始线开始，可以移动，旋转，调整大小直到位置，方位，长度满足要求。

1. 初始化线工具

在应用线工具前，首先设定端点为合适值。可以用输入坐标或者用鼠标选择。线工具激活后如下图：

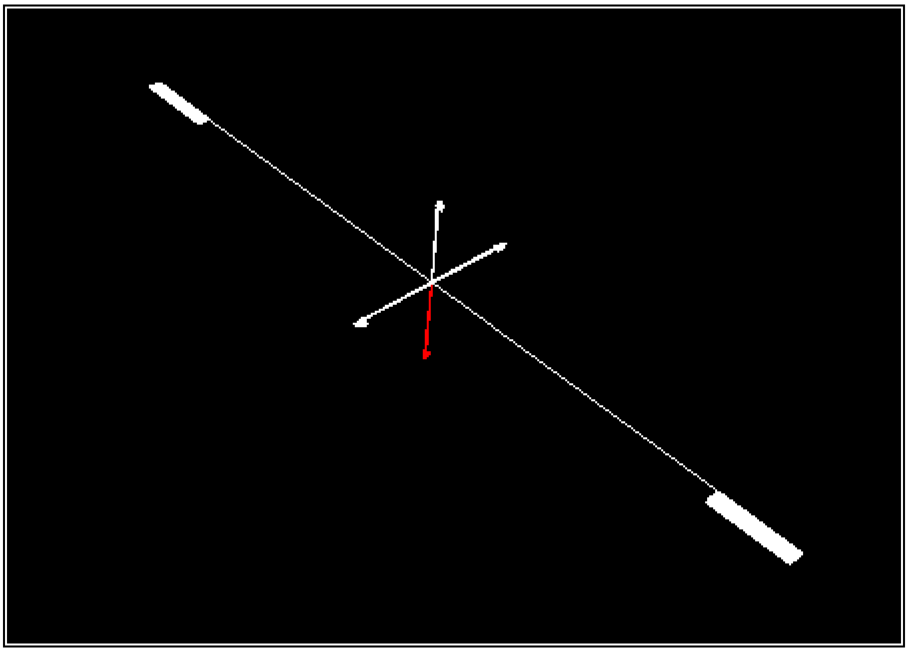


图 11-5-2：线工具

接下来就可以方便的旋转，移动，和调整大小。

2．移动线工具

在任何位置点击鼠标探测键便可以沿着红轴线方向移动线。绿色箭头指示运动的方向。可以通过 shift 键转换移动方向。

3．旋转线工具

需要用鼠标的探测键点击有白色箭头的轴。在点击时，会出现包含其它有箭头轴的绿色的带子，它作为旋转的轴线。拖动鼠标，便可旋转线工具。

4．调整线工具尺寸

若希望生成耙表面，可以通过调整线工具来改变耙面的长度。用鼠标的探测按钮点击线工具端点的白色矩形，并拖动鼠标来放大或缩小工具。绿色的箭头指示缩放方向。

5．重新设定线工具

在线工具丢失时，可以用面板的 reset 按钮或者先关闭线工具再重新激活。

11.6 平面

在 3D 问题中可以创建任意方向的流域平面显示计算数据。共可以创建六种类型的平面：

1．与流域无限大平面的交界面。

2. 边界平面，为平行六面体平面，数据点不等距。
3. 等间距数据点边界平面：这种平面与上述不等间距平面大体相同，不同之处在于必须指定两个方向的数据点密度。
4. 通过指定点定法线方向平面：要创建该种类型平面首先定义法线矢量方向和一个数据点。
5. 与现存面对齐的平面：创建这种平面的输入条件为一个平行平面和一个指定点。
6. 与图形窗口对齐的平面：要创建者这种平面的输入条件为，定义一个独立的点。

创建平面的步骤如下：

1.打开平面面板

Surface -〉Plane...

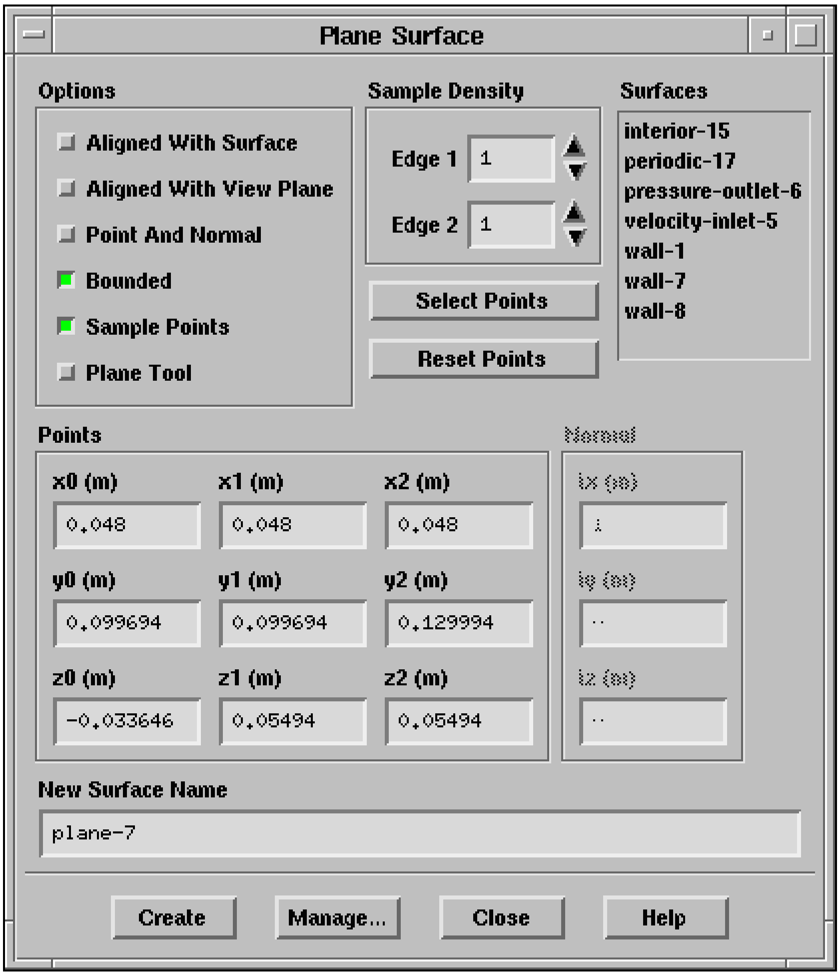


图 11-6-1 平面面板

1. 首先在面板中选择生成平面类型。
2. 指定平面位置，有三种方法：
1）输入坐标
2）用鼠标，首先点击 select points 按钮，然后在图形窗口中用鼠标探测键选择三个点。
3）用平面工具。

3. 可以选择输入新名称。
4. 点击 creat 按钮。

11.6.1 应用平面工具

应用平面工具可以互动的精确定义平面。从起始点出发，应用平面工具可以方便的移动，旋转，调整平面的大小到预期的位置，方向，大小等。

1. 初始化平面工具

在打开平面工具前，必须首先设定点为合适的初始值。可以手动输入值也可以用 select points 按钮。

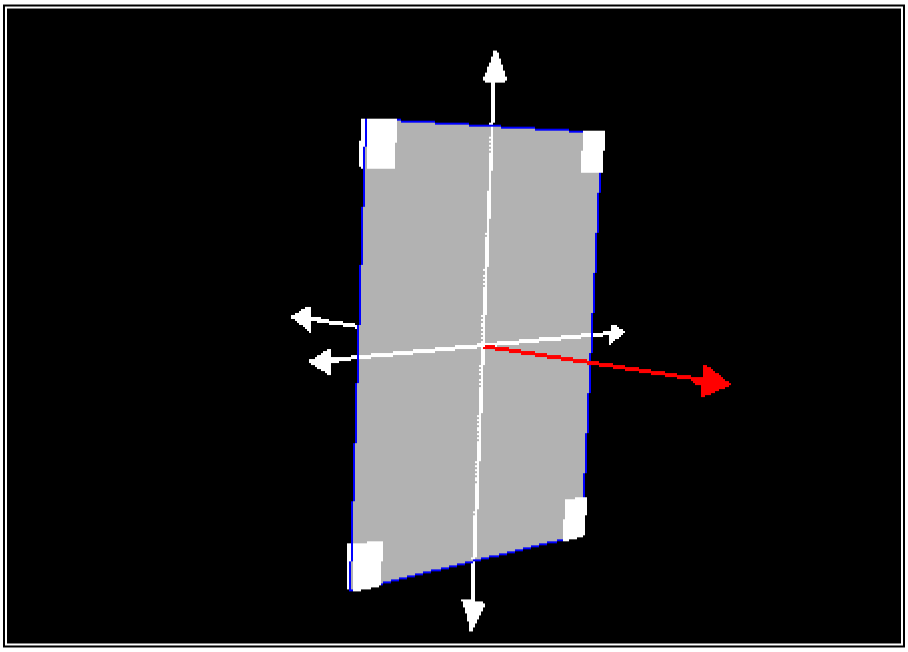


图 11-6-2 平面工具

2. 移动平面工具

要沿着平面地法线方向移动平面工具，可以在平面工具的任何灰色部分点击鼠标的探测键然后拖动直到满意为止。绿色箭头标志移动方向。要沿着横向移动平面，按下 shift 键，用鼠标的探测键点击平面工具中的任意灰色部分，然后就可沿横向拖动平面。绿色的两套箭头表示可能的移动方向。

3. 旋转平面工具

要旋转平面工具，需要用鼠标的探测键点击平面工具轴线的白色箭头。点击一个方向的箭头，便可以沿着另外两个方向中的任意一个旋转。在点击后会出现两条包围着平面工具的绿色带子，当旋转平面时，有旋转方向的带子会变成黄色。

有以下小技巧可以帮助旋转平面工具：

（1） 一旦眼某个方向旋转，除非释放按键，否则不能改变旋转方向。
（2） 不要点击红色箭头进行旋转。
（3） 不要点击远离你的箭头进行旋转，因为这样很难判断旋转的方向。
（4） 不要一次旋转超高 90 度。

4. 调整尺寸

如果计划生成边界平面，可以调整平面的尺寸来确定边界。用鼠标的探测按钮点击平面工具的白色角部，可以重新调整大小。

5. 重新设定平面工具

若丢失平面工具，可以用 reset points 或者重新初始化平面工具。

11.7 二次曲线表面

可以用输入而此曲线方程系数的方法来确定二次曲面。有线、面、圆、球以及其他二次曲线形式。步骤如下：

Surface －〉Quadric...



图 11-7-1 二次曲线面板

1. 确定二次曲面形式。
2. 按照国际单位输入定义方程：
1）线和平面：对于线或平面方程如下：

Ix*x+iy*y+iz*z=distance，ix，iy，iz 分别为 x，y，z 的系数。
2）对于球面，方程为：
$(x-x0)^2+(y-y0)^2+(z-z0)^2=r^2$
需要输入 x0，y0，z0，和 r^2。
3）二次表面，需要输入 x^2，y^2，z^2，xy，yz，zx，x，y，z 的系数。
3. 可以选择输入新名称。
4. 点击创建按钮。

11.8 等值面

可以根据 x，y，z 坐标轴创建等值面，会显示 x，y，z 轴横截面的等值面。创建等值面的步骤如下：

Surface->iso-surface

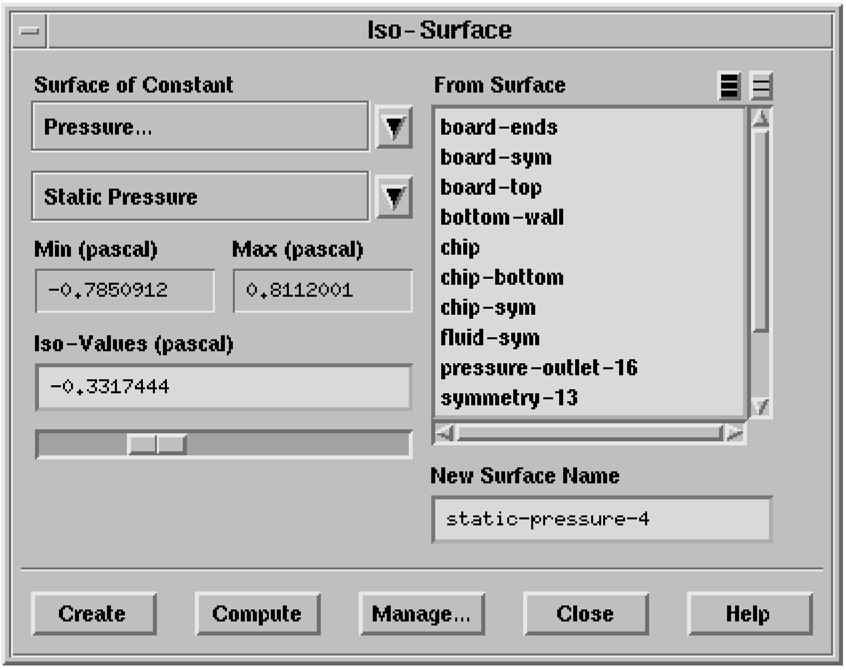


图 11-8-1 等值面面板

1. 选择标量。
2. 如果希望在已存的面上建立面则选择该面，否则面会生成在整个流域中。
3. 点击 compute 按钮获得最大最小值。
4. 设定数值，可以由下面两种方法：
1）用滑动条选择；
2）直接输入数值
5. 选择输入新名字。
6. 点击创建按钮。

11.9 折叠表面

如果已经创建了一个表面，但是不希望用整个表面显示数据，这时可以用 clip 把面裁减为两个值之间的剪切面。剪切面包含数值在两个标量范围之间的数据点。例如，对于图 11.9.1 中的外壁面进行了和 x 轴与 0 之间的剪切，结果是只显示外壁面的背部，这样就可以清楚地看到从吸入口进入的值。。

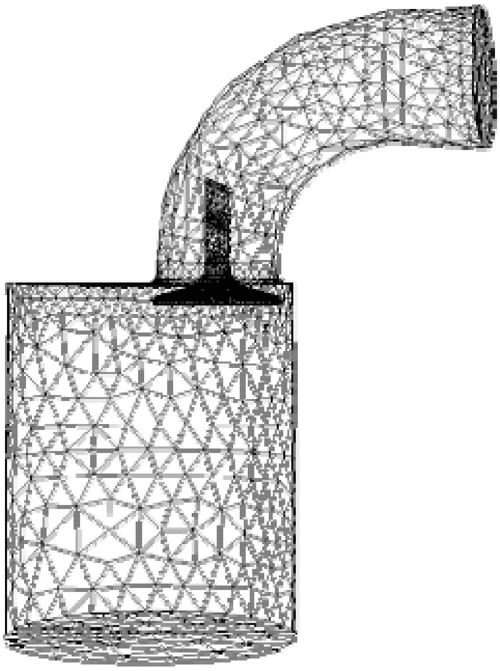


图 11-9-1 外壁面对 x 轴数值进行剪切

要剪切一个表面，步骤如下：

1.　首先打开等值剪切面板：

Surface->iso-clip

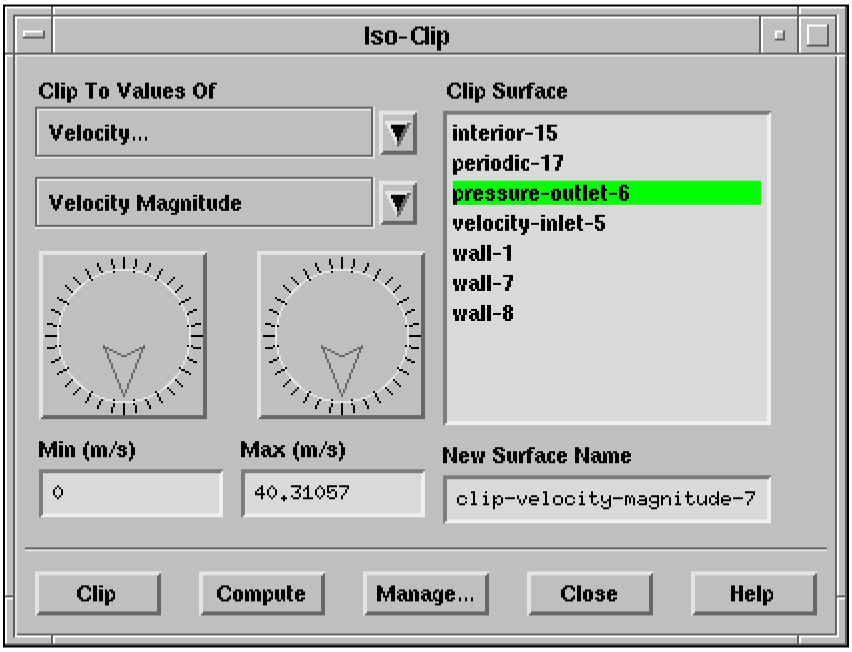


图 11-9-2 iso-clip 面板

2. 选择标量。
3. 选择表面。
4. 点击 compute 按钮，来确定最大最小值。
5. 定义剪切范围。
1）用数据表盘选择。
2）手动输入数值。
5. 可以选择新名称
6. 点击 clip 按钮

11.10 改变表面形状

可以通过利用已有的表面进行旋转移动等操作而创建新的表面。例如，可以旋转复杂的叶轮机械的表面来显示叶面之间的数据。

对现有表面进行变形操作的步骤如下：

1. 在变形面板中

Surface->transform

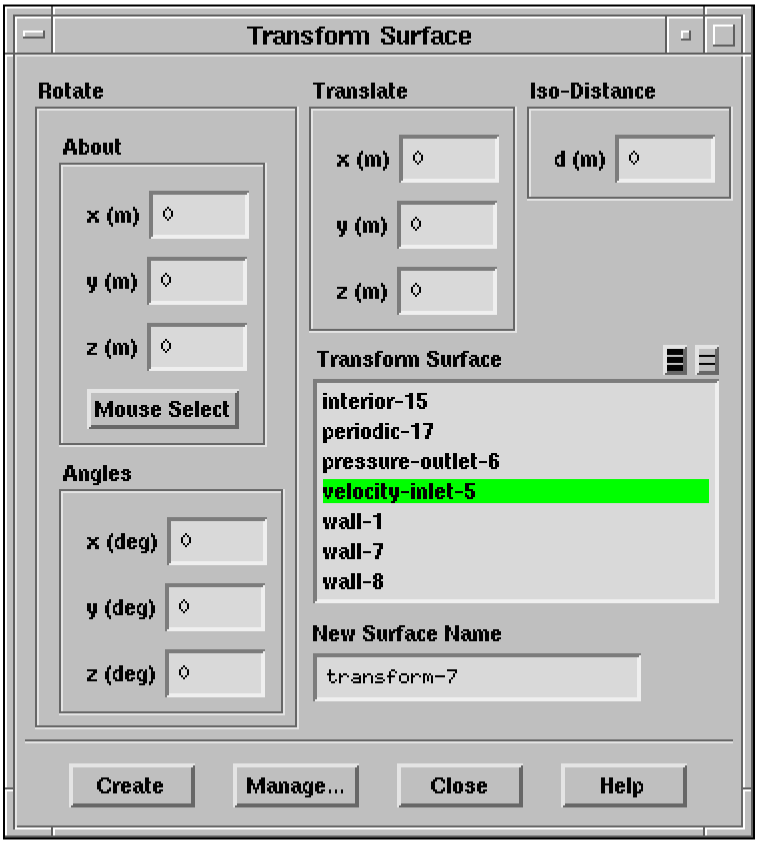


图 11-10-1 表面变形面板

1. 选择已有表面。
2. 设定变形参数，共有移动，旋转，等间距三种操作。

1） 旋转，需要指定旋转方向，起始点和角度。

在 rotate 项下面的 about 对话框中，需要确定一个点。例如，如果输入（1，5，3）会沿着坐落在（1，5，3）点的 x,y,z 坐标系进行旋转。可以输入坐标或者用鼠标在图形窗口中选择。

在 rotate 下面的 angels 中，输入要旋转的角度。

2）移动：移动相对简单，只要确定距离和方向就可以了。

3）等值化：要创建距离初始面固定垂直距离的表面，只要确定垂直距离和变形的表面就可以了。

3. 可以选择输入新名字。
4. 点击 creat 按钮。

11.11 分组、改名、删除表面

首先打开 surfaces 面板。

Surface->manage

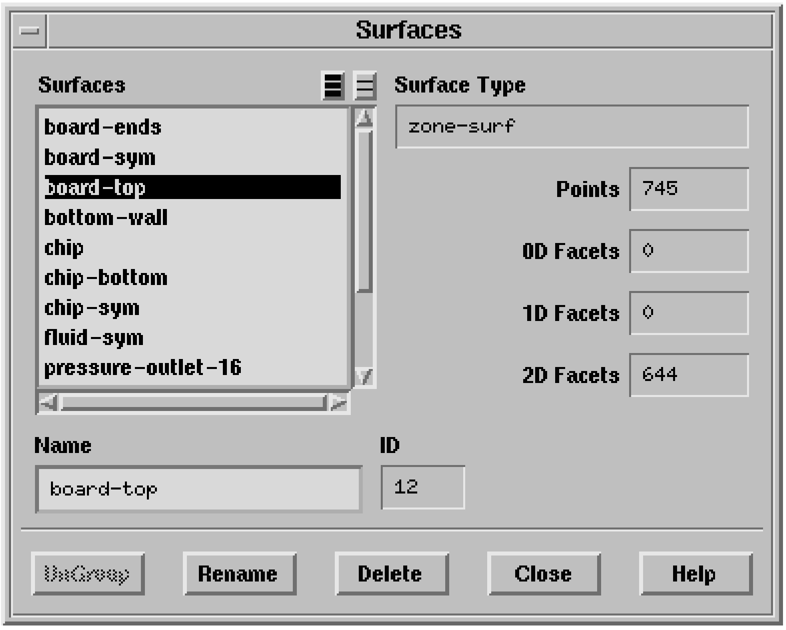


图 11-11-1 表面面板

1. 表面分组

可以把几个表面作为一个组来方便的一起进行后处理操作。要创建一个表面组，首先在列表中选择，然后点击 group 按钮就可以了。要解散组，先选择组再点击 ungroup 按钮就可以了。

2. 重命名表面

重命名非常简单，只要在列表中选择该表面，然后输入新名字，再点击 Rename 按钮就可以了。

3. 删除表面

对于没有用处的表面，可以首先在 surface 列表中选择该表面，然后按 delete 按钮进行删除。

4. 表面统计

用表面面板，还可以读取表面的特征信息。Points 是表面上共有的点数。0DFacet 是面上单独点的个数，1Dfacets 是面上包含两个相连点面的个数，2Dfacets 为 3D 形状中 2D 面的个数。这里要注意在 3D 问题中 interior 包含 2Dfacet,而 2D 问题中包含 1Dfacet.

第 12 章 图形及可视化技术

利用 FLUENT 提供的图形工具可以很方便的观察 CFD 求解结果，并得到满意的数据和图形，用来定性或者定量研究整个计算。本章将重点介绍如何使用这些工具来观察您的计算结果。本章包含如下内容：

1. 基本图形生成
2. 调整图形显示方式
3. 控制鼠标功能
4. 改变观察方式
5. 画一个场景
6. 动画技术
7. 生成视频
8. 历史曲线与 XY 插值曲线

12.1 生成基本图形

在 FLUENT 中能够方便的生成网格图、等值线图、剖面图，速度矢量图和迹线图等图形来观察计算结果。下面将介绍如何产生这些图形。

12.1.1 生成网格图

在问题求解的开始或计算完成需要检查计算结果时，往往需要能够观察某些特定表面上的网格划分情况。如图 12-1-1 所示，在 Fluent 中可以利用强大的图形功能显示求解对象的部分和全部轮廓；图 12-1-2 所示为网格线（边）；图 12-1-3 所示为一个三维计算对象的网格实体表面；图 12-1-4 为计算对象表面上的网格节点图。

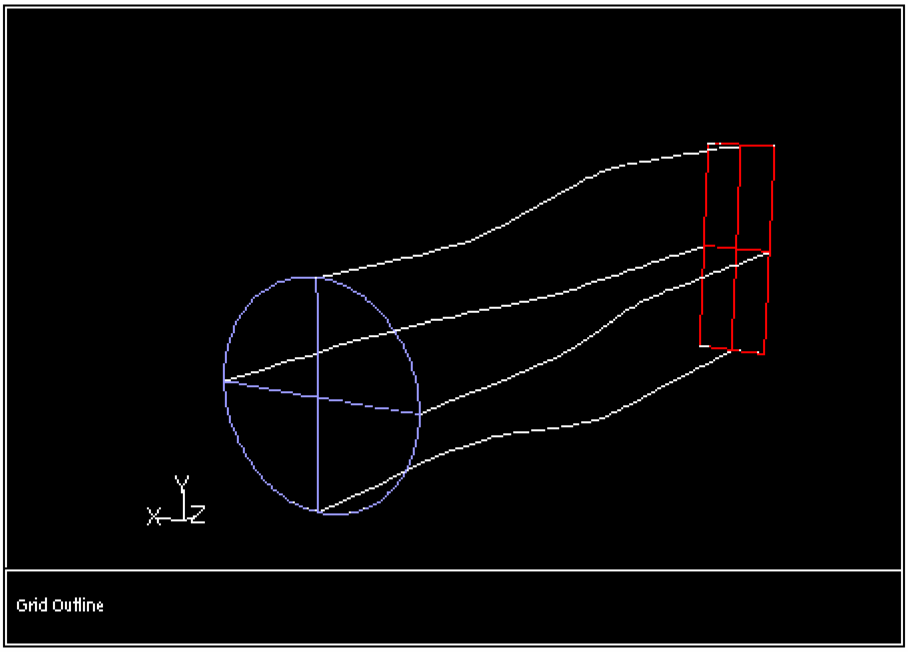


图 12-1-1：显示轮廓

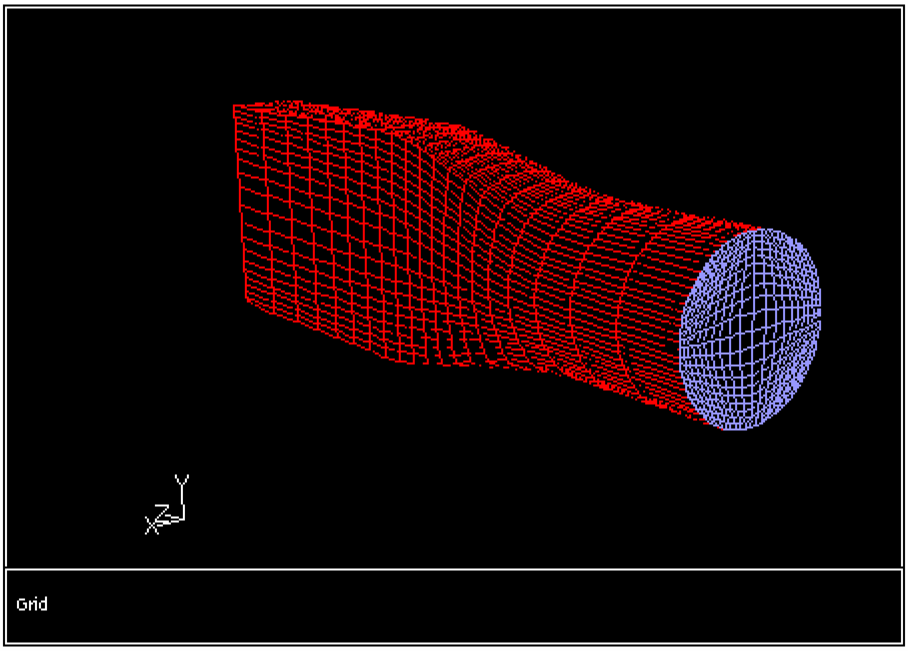


图 12-1-2：网格线的显示

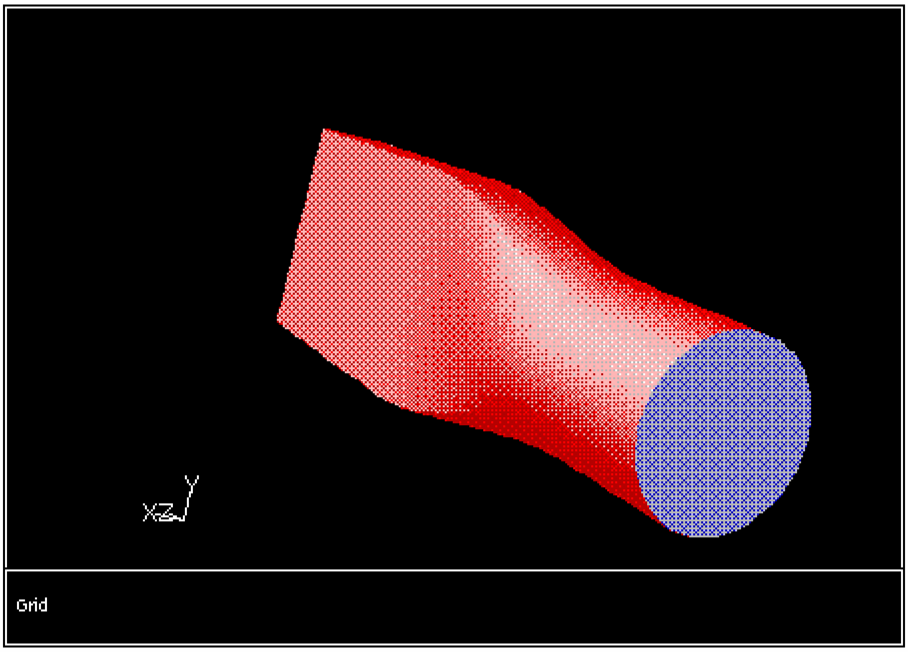


图 12-1-3：网格面（网格填充）显示

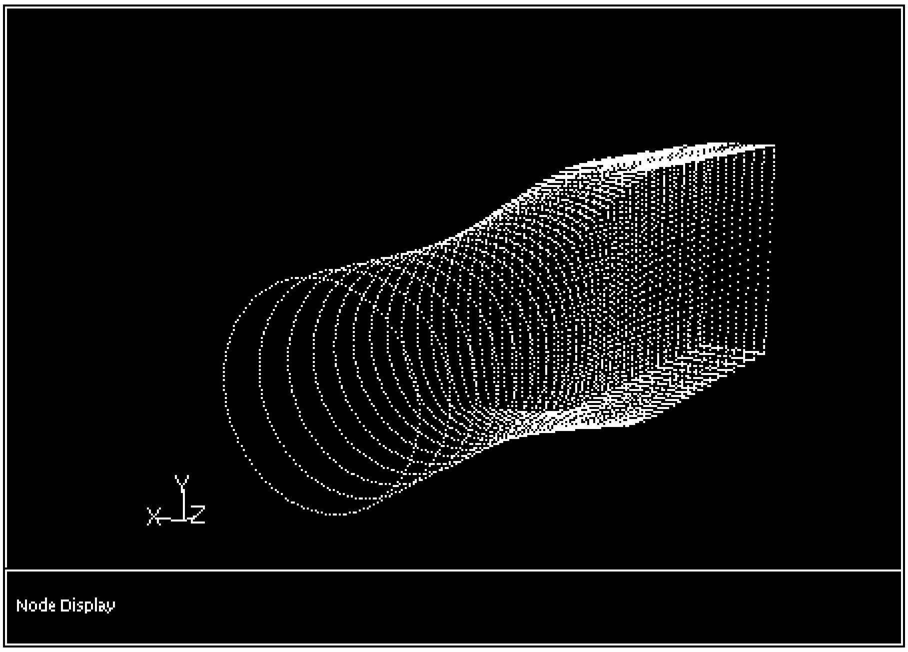


图 12-1-4：网格节点显示

2．生成网格或轮廓线视图的步骤

（1）打开网格显示面板

菜单：Display –〉Grid...

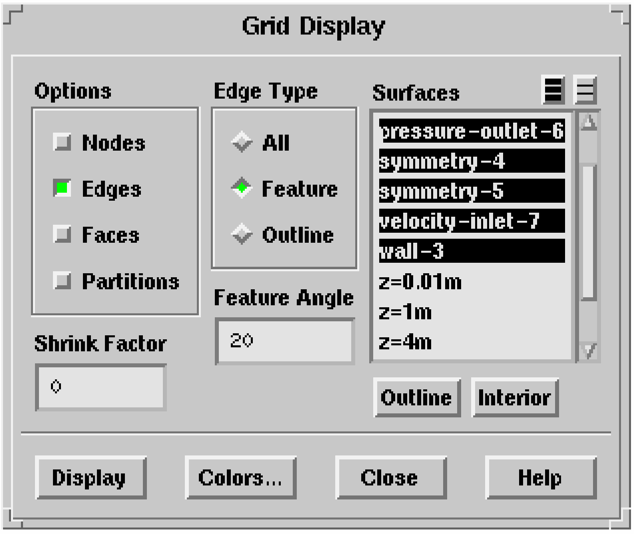


图 12-1-5: 网格显示对话框

（2） 在表面列表中选取表面。点击表面列表下的 Outline 按钮来选择所有“外”表面。如果所有的外表面都已经处于选中状态，单击该按钮将使所有外表面处于未选中的状态。点击表面列表下的 Interior 按钮来选择所有“内”表面。同样，如果所有的内表面都已经处于选中状态，单击该按钮将使所有内表面处于未选中的状态。

（3） 根据需要显示的内容，可以选择进行下列步骤：

1）显示所选表面的轮廓线，在图 12-1-5 所示的对话框中进行如下设置：在 Options 项选择 Edges，在 Edge Type 中选择 Outline。

2）显示网格线 (如图 12-1-2), 在 Options 选择 Edges，在 Edge Type 中选择 ALL。

3）绘制一个网格填充图形 (如图 12-1-3), 在 Options 选择 Faces。

4）显示选中面的网格节点 (如图 12-1-4), 在 Options 选择 Nodes。

（4）设置网格和轮廓线显示中的其它选项。

（5）单击 Display 按钮，就可以在激活的图形窗口中绘制选定的网格和轮廓线。

如果选择绘制网格填充图形, 并且希望图形光滑，这时需要打开光源（类似于一般 3D 软件中的光源设置），并选择一种光线插值方法而不是 Flat 选项。

如果您显示网格节点, 并希望通过指定符号来代替节点, 您可以通过在显示选项对话框中设置 Point Symbol 来实现。

注意, 对于具备周期性边界条件的求解问题, 您在求解对象的两侧均不能看到其边界。为了在网格的显示中包含周期性“阴影”边界, 可以使用 display/grid 命令。

3．网格和轮廓显示选项

该选项在生成网格或轮廓线步骤的第三步中曾经提到过，它包含了修改网格颜色，显示增加轮廓线等重要特征，例如绘制分割边界，以及在显示中缩小面或单元。

FLUENT 允许您通过管理颜色对每一个区域类型或表面来显示具有不同颜色的网格。这个特性可以帮您迅速方便的理解网格绘制。如果需要修改颜色，点击网格显示对话框上的 Color 按钮来打开网格颜色对话框（图 12-1-6）

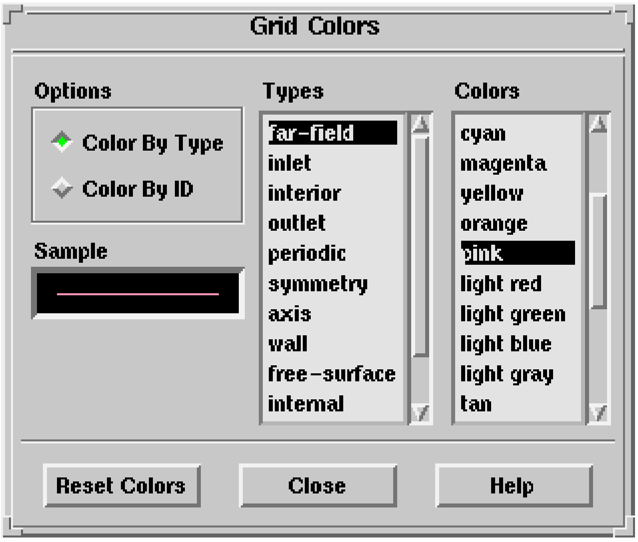


图 12-1-6: 网格颜色对话框

（可以通过场景描述对话框开为每一个需要显示网格的平面单独设置颜色）

如果要改变一个指定区域类型的网格的显示颜色，从类型列表中选定区域类型并在颜色列表中选择新的颜色。当再次显示网格时，会看到颜色改变后的效果。注意 Types 列表中的表面类型适用于全部的表面网格（例如，使用 Surface menu 打开面板创造的表面的网格是画好的）除了表面区域。

对于封闭的 3D 物体如柱体，标准轮廓线显示通常不能精确的显示其细节。这是因为对于每一个边界，只有这些在几何体外侧的边(例如一些在边界上只被一个表面使用的边)才被显示。图 12-1-7 为一个复杂的管道的轮廓线，只有入口和出口是可见的。

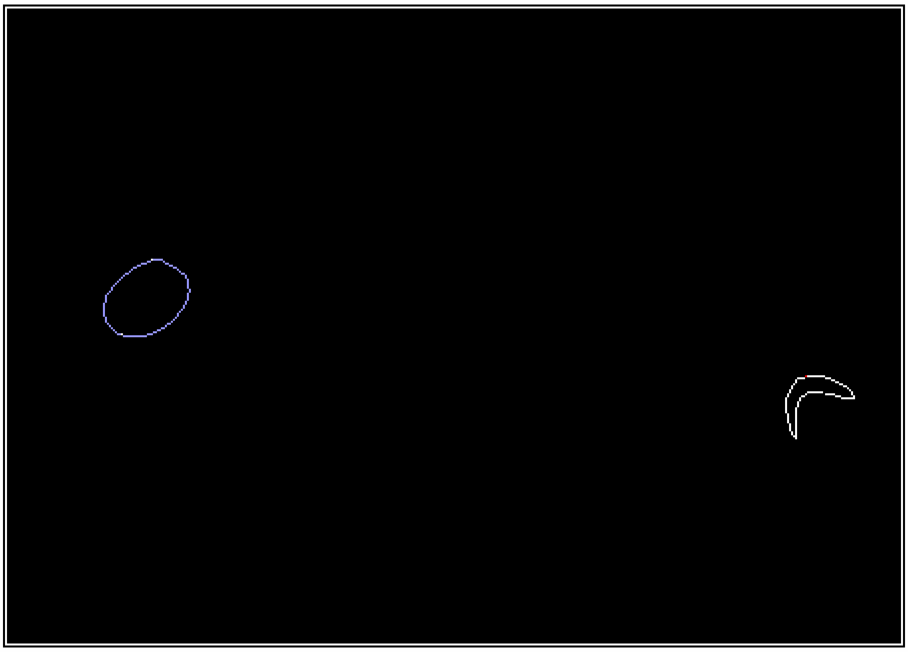


图 12-1-7: 复杂管道标准轮廓线

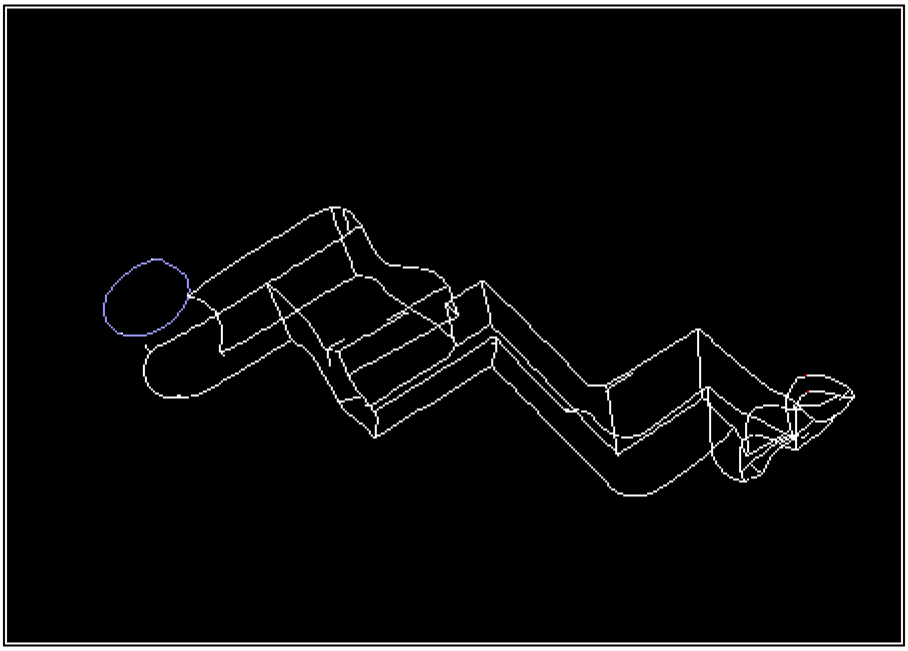


图 12-1-8:复杂管道特征轮廓线

可以通过网格显示对话框中的特征选项得到额外的轮廓线特征（如图 12-1-8）。在 Edge Type 中选择 Feature 选项，然后设置特征角度。默认特征角度为 20 度， 如果两个邻接表面法向方向的夹角大于 20 度，那么在这两个面中的边都将被绘制。减小特征角度可以显示更多的边（细节）能够在轮廓线的绘制中显。角度的大小取决与几何体的曲率和复杂度，可以采用逐渐修改特征角度的策略直到得到满意的轮廓线。

在处理并行处理分割网格时，可以通过在网格显示对话框中选择 Partitions 选项将分隔的边界增加到网格显示中。

如果需要在网格显示中区别个别的面或单元，或者希望扩大两个邻接面或单元间的距离，可以通过在网格显示对话框中增大 Shrink 系数实现。采用默认值 0，会使得在显示中邻接的面或单元的边发生重叠。而当值为 1 时，会出现相反的情况：每一个面或单元都被一个点代替，而其之间存在相当大的距离。一个很小的值如 0.01 就可以将邻接的面或单元区分开来。采用不同值的收缩系数显示效果如图 12-1-9，图 12-1-10 所示。在改变收缩系数后必须再次点击 Display 按钮才能观察到其效果。

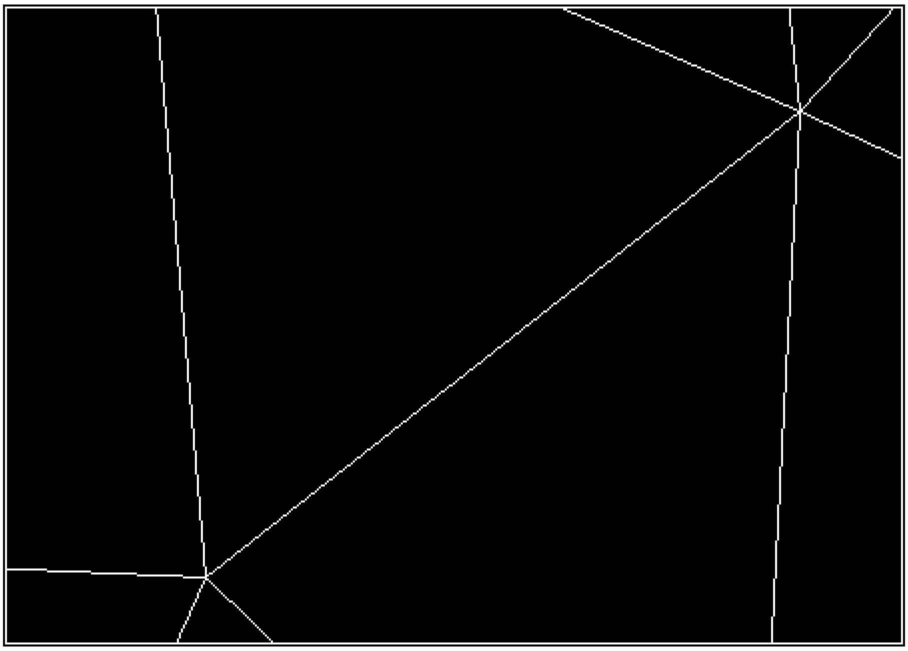


图 12-1-9: 收缩系数为 0 的网格显示

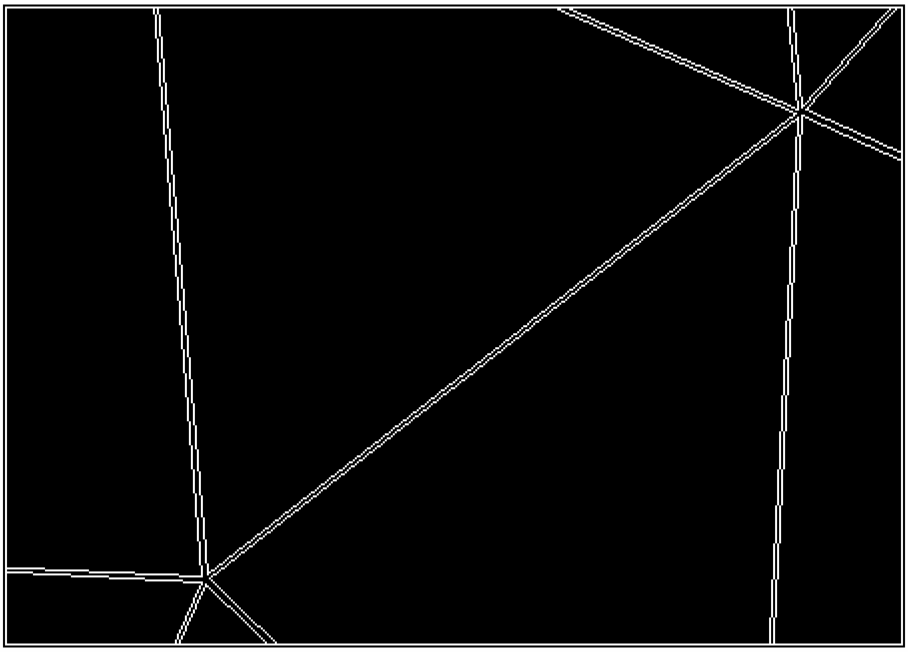


图 12-1-10: 收缩系数为 0.01 的网格显示

12.1.2 绘制等值线和轮廓图

FLUENT 中可以在求解对象上绘制等值线或外形的。等值线是由某个选定变量（如等温线，等压线）为固定值的线所组成。而轮廓则是将等值线沿一个参考向量并按照一定比例投影到某个面上形成的。图 12-1-11 和图 12-1-12 分别为等值线和轮廓的例子。

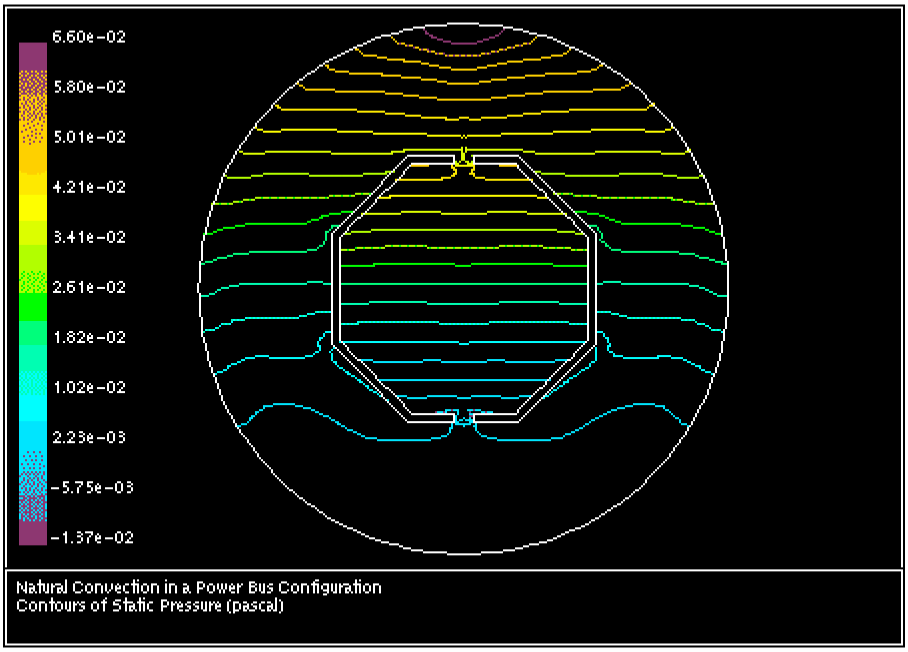


图 12-1-11：静态压力等值线

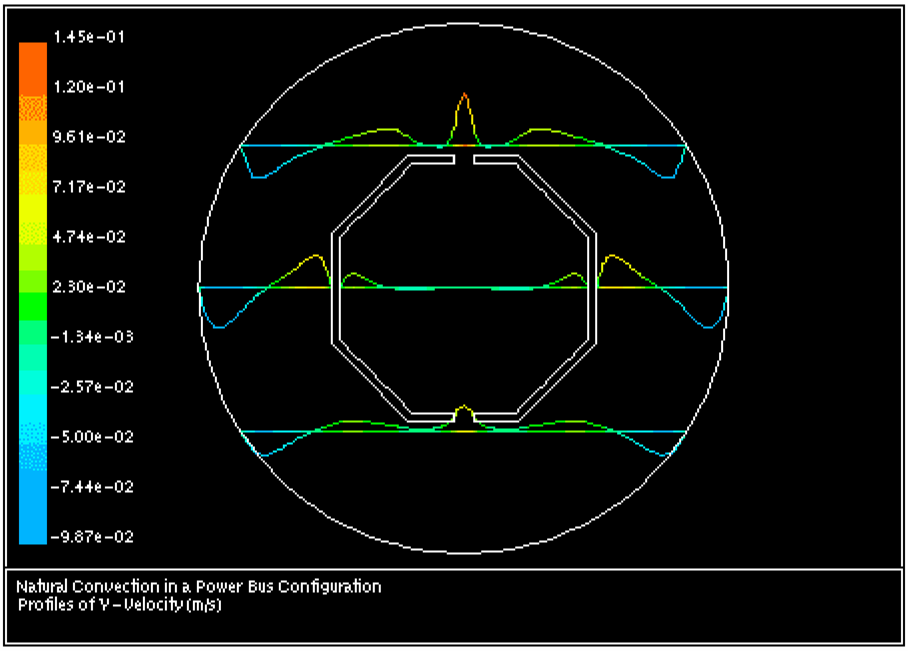


图 12-1-12:Y 向速度轮廓

1．生成等值线和轮廓的步骤

通过图 12-1-13 所示的等值线对话框来生成等值线和轮廓。

菜单：Display –〉Contours...

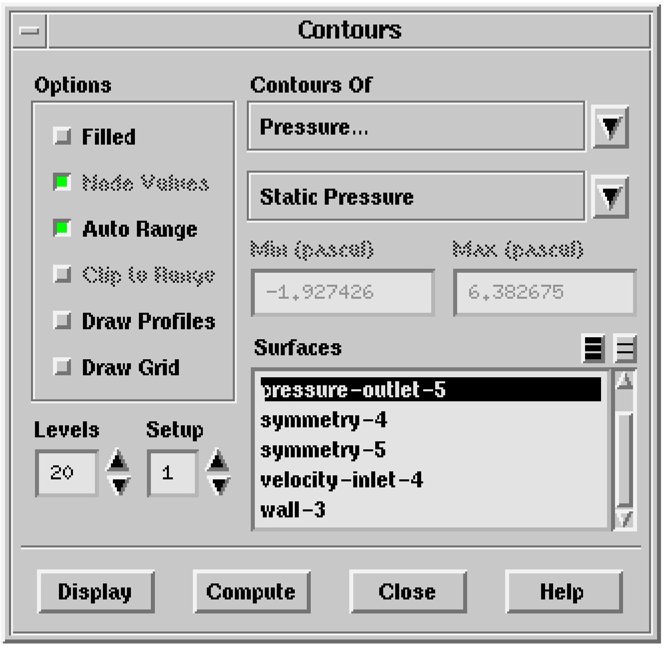


图 12-1-13： 等值线对话框

生成等值线或轮廓的基本步骤如下：

（1） 在 Contours Of 下拉列表框中选择一个变量或函数作为绘制的对象。首先在上面的列表中选择相关分类；然后在下面的列表中选择相关变量。

（2） 在 Surfaces 列表中选择待绘制等值线或轮廓的平面。对于 2D 情况，如果没有选取任何面，则会在整个求解对象上绘制等值线或轮廓。对于 3D 情况，至少需要选择一个表面。

（3） 在 Levels 编辑框中指定轮廓或等值线的数目。最大数为 100。

（4） 如果需要生成一个轮廓视图，请在 Option 中选中 Draw Profiles 选项。在轮廓选项对话框中（如图 12-2-4），可以如下定义轮廓：

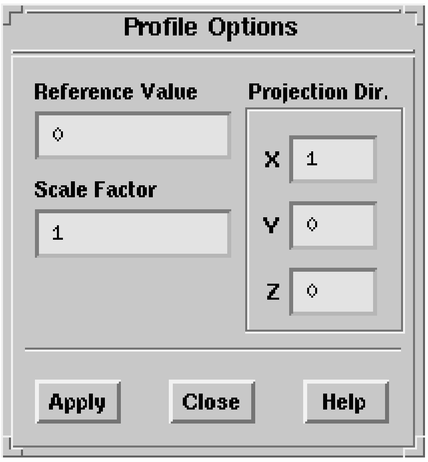


图 12-1-14: 轮廓选项对话框

1) 在 Reference Value 中为轮廓设置“0 高度”参考值，并在比例系数（Scale Factor）中设置投影的长度比例因子。在定义的面上，任何节点值等于参考值（Reference Value）点都将被绘制在轮廓上。大于参考值（Reference Value）的点将被投影到定义面的前面。（按照投影方向（Projection Dir）中定义的方向）并且根据比例系数（Scale Factor）中的值进行缩放；小于参考值（Reference Value）中的数值的点将被投影到定义面的后面

并进行缩放。

当需要显示一个变量的变化时，如果变化值和变量值相比较小，采用比例系数参数可以用来产生较全面的轮廓。例如，当需要显示温度变化范围为 300K 到 310K 的温度轮廓时，如果我们采用默认的比例系数（以绝对温度 310K 为基数）绘制温度轮廓，那么 10K 的变化在图中很难显示出来；为了产生一个较完善的轮廓，可以将参考值（Reference Value）设置为 300 并设置比例系数（Scaling Factor） 为 5 来放大 10K 范围在轮廓上的显示大小。这样就能够详细的观察到 10K 范围的变化。

在随后的轮廓显示中，温度为 300K 位处于基准线，而其它数据在显示时，首先将减去 300，并将其差值绘制在轮廓上。因而，轮廓上的图像只显示了相对于 300K 的温度变化。

2) 设置轮廓的投影方向(Projection Dir.)。如果是 2D 图形，例如，在求解对象上的压力等值线可以按 z 方向进行投影来形成一个烟丝图（carpet plot），或者对 y 方向速度等值线沿 y 方向的切片进行投影形成一系列的速度轮廓图（例如图 2 的等值线和轮廓图）

（3）点击 Apply 按钮关闭轮廓选项对话框。

（4）设置等值线和轮廓选项对话框中的其它选项。

（5）点击显示（Display）按钮在激活的图形窗口中来绘制指定的等值线和轮廓。

显示的结果将包含选定变量的指定的等值线和轮廓的指定数目，同时并将其值量级的变化范围在最小和最大区域按照增加的方式进行显示。

2．等值线和轮廓绘制选项

在生成等值线和轮廓步骤中的第五步提到过该选项，包含了绘制用颜色填充的等值线/轮廓线（替代默认的等值线/轮廓），指定待绘制等值线轮廓变量的范围，也包括在等值线轮廓中网格的部分显示，选择节点或单元的值进行显示，以及存贮等值线轮廓相关设置。

色彩填充等值线或轮廓图是用连续色彩显示的等值线或轮廓图形显示（如图 12-2-5），而不是仅仅使用线条来代表指定的值。（注意一个色彩填充的轮廓显示通常被当作“烟丝图”）。可以在生成等值线和轮廓的第五步选中等值线对话框中的充填（Filled）选项来绘制一个色彩充填的等值线或轮廓图。

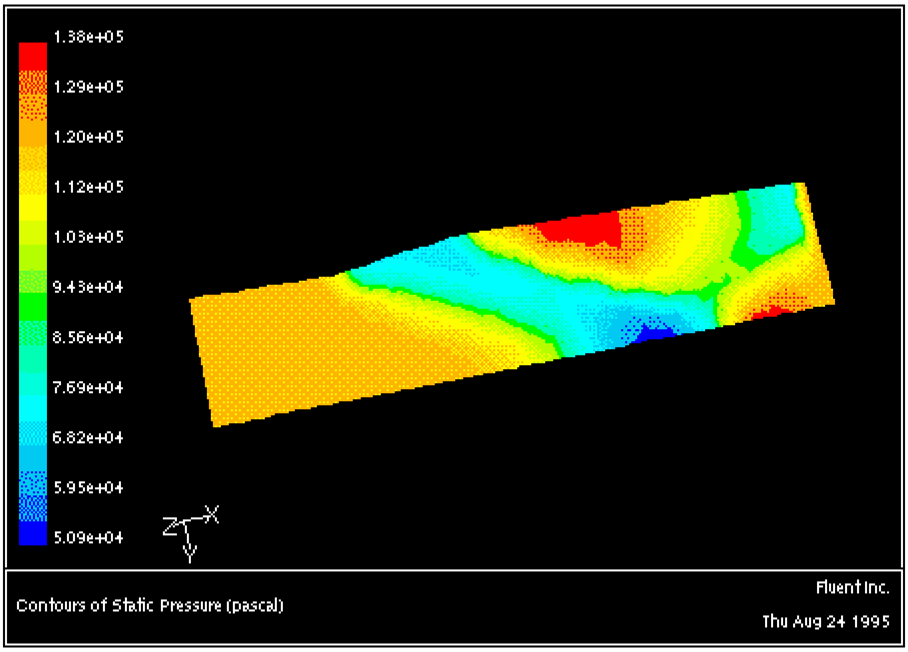


图 12-1-15：色彩充填的静态压力等值线图

为了光滑显示效果，可以通过调整光源完成，并选择一个适合的光线插值方法来完成，而不是在显示选项对话框或光线对话框中进行相关设置。 请注意如果 Clip to Range 选项一旦被选中，不会得到光滑图形。

默认情况下，等值线或轮廓的变化范围通常被设置在求解对象结果的变化范围内。这意味着在求解对象内色彩变化将以最小值（显示在 Min 区域）开始，以最大值（显示在 Max 区域）结束。如果所绘制的等值线或轮廓只是求解对象的一个子集（例如一个表面），绘制结果可能只覆盖色彩变化的一部分。例如，假设用蓝色代表 0，用红色代表 10，而绘制的表面变化只是从 4 到 6，由于绿色是由蓝到红的中间色彩，因而所绘制的图像上很可能是绿色的等值线或轮廓。如果所关心的值在一个小范围内变化，如蓝色代表 4 而红色代表 6，可以自行设置显示范围。如果只关心特定的值，也可以自行设定显示范围，例如，如果想了解哪个区域的压力超过了特定的值，可以增加显示的最小值，压力低于该值的不再被显示。

如果希望自行设置等值线/轮廓的显示范围，首先使等值线对话框中的 Auto Range 选项处于实效状态。Min 和 Max 编辑框则处于可编辑状态，然后可以输入显示的范围。在显示默认范围时，点击 Compute 按钮将更新 Min 和 Max 的值

在绘制色彩填充等值线/轮廓（如前在等值线轮廓选项中描述），可以控制值超过显示范围的值是否显示。Clip to Range 选项的默认状态为选中，这使得超出显示范围的值不被显示（例如不绘制其等值线/轮廓）。但是如果您使其处于实效状态，低于 Min 的值将会以代表最低值的色彩进行显示，而高于 Max 的值将以代表最高值的色彩进行显示。图 2 和图 3 则分别显示了选中/未选中 Clip to Range 选项时的绘制结果。

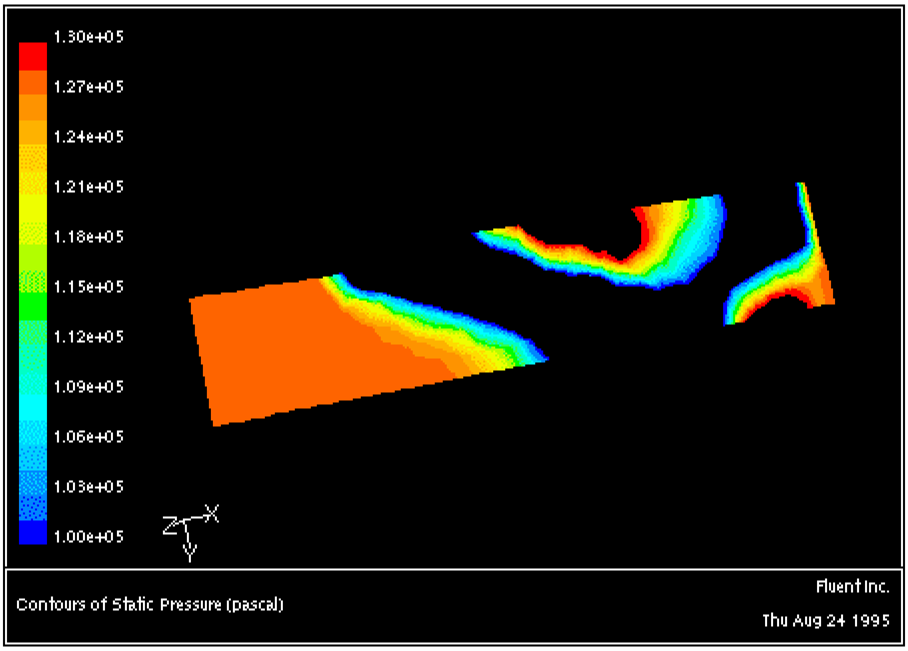


图 12-1-16: 选中 Clip to Range 选项时的绘制结果

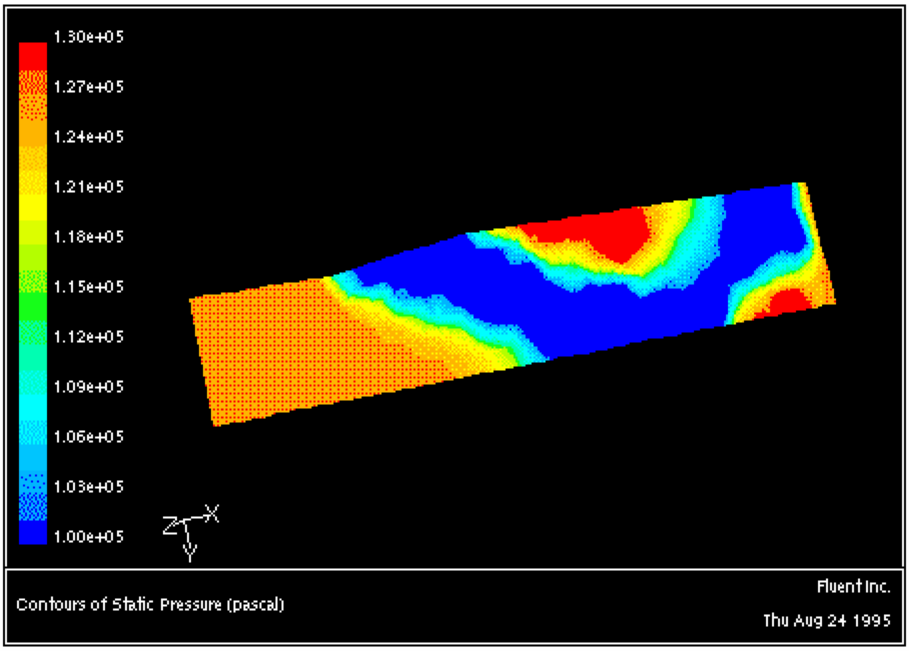


图 12-1-17: 未选中 Clip to Range 选项时的绘制结果

对于某些要求解的问题，尤其时 3D 几何体，可能希望在等值线/轮廓中包含部分网格作为空间参考点。例如希望在等值线中显示入口和出口的位置。这可以通过选中等值线对话框中的 Draw Grid 选项完成。当选中 Draw Grid 选项后，显示对话框将会出现，在这个对话框中可以设置网格显示参数。当点击等值线对话框中的显示（Display）按钮，在等值线/轮廓中将会显示出按照在网格显示对话框中定义的网格。

在 Fluent 中，可以选择显示计算得到的单元节点中心的值或者按照插值计算的节点的值。默认情况下，节点值（Node Values）选项被选中，插值计算的节点值被显示。对于等值线或轮廓，总是采用节点的值。如果要显示色彩填充的等值线或轮廓，最好采用单元节点中心值，使用该值只需将 Node Value 选项关闭。采用节点值绘制色彩填充等值线/轮廓将按颜色的层次进行光滑显示，而使用单元节点中心值绘制的色彩填充的等值线/轮廓则会显示出一个单元到其邻接单元颜色的显著变化。

如果需要绘制一个多孔或扇叶的显示图像，来描述一个脉冲或其它不连续或跳跃的变量，应该采用单元节点中心值，如果在该情况下使用节点值，不连续效果将会由于节点的平均而不会在图像中清晰的显示。

对于频繁使用的等值线变量和选项，可以在等值线对话框中指定一个 Setup number，并设定所需信息以此来生成等值线图形。当单击 Display 按钮，所设定的 Contours Of, Min, Max 以及表面参数将被存储。可以将 Setup number 设置成一个未被使用的值（如，一个没有存储任何信息的 ID 号）然后生成一个不同的等值线图形。要按存储的信息显示图形，只需将 Setup number 重新设置成保存信息的变化，然后单击 Display 即可。可以保存 10 个不同的编号。

！注意等值线层的数目，在网格显示对话框（当 Draw Grid 选项被激活）中选定要显示的表面，以及在轮廓选项对话框（当 Draw Profiles 选项被激活）中对轮廓的设定均不会在 Setup 中存储，同时 Setups 也不会被存储到工作文件中。

12.1.3 绘制速度向量图

除了等值线图与轮廓图，另一种经常用到的结果处理图为在或选中的表面上绘制速度向量图。默认情况下， 速度向量被绘制在每个单元的中心（或在每个选中表面的中心），用长度和箭头的颜色代表其梯度（图 12-3-1）。包括几个向量绘制设置参数，可以用来修改箭头的间隔、尺寸和颜色。注意在绘制速度向量时总是采用单元节点中心值；不能采用节点平均值进行绘制。

.

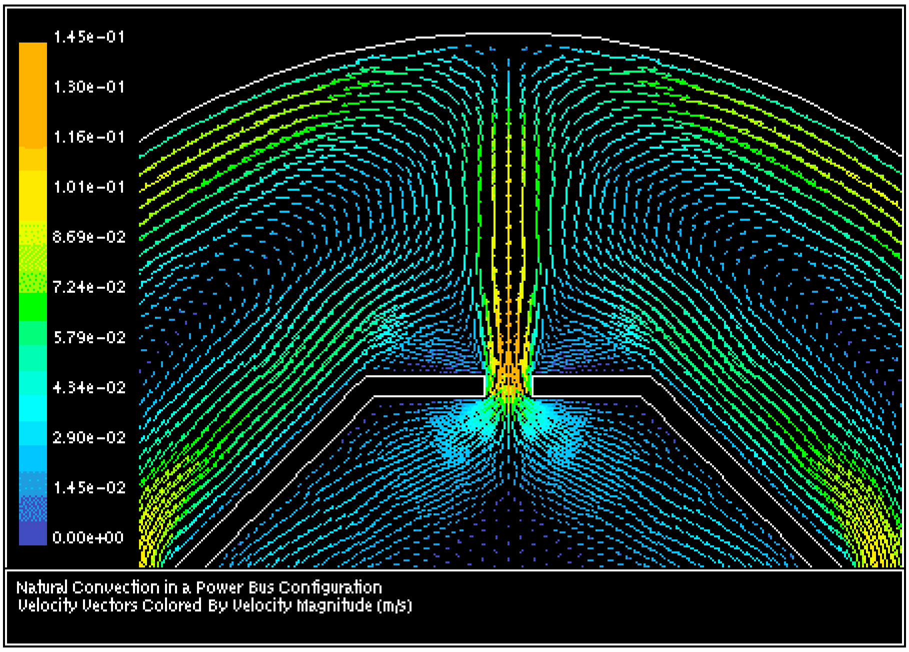


图 12-1-18: 速度向量图

1．生成速度向量图的步骤

（1）可以通过速度向量对话框来生成速度向量图 (如图 12-3-2).

菜单 Display –〉Velocity Vectors...

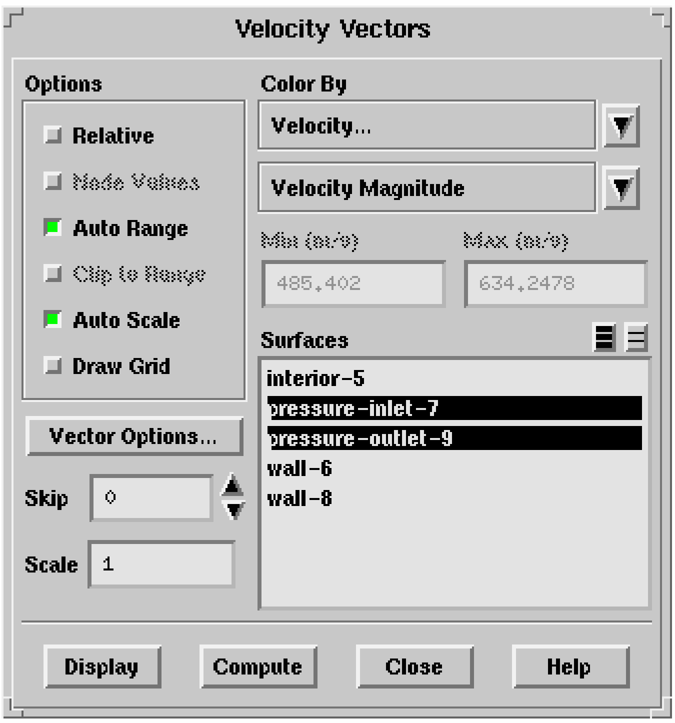


图 12-1-19: 速度向量对话框

（2）在 Surfaces 列表中，选择希望绘制其速度向量图的表面。如果希望显示的对象为整个求解对象，不要选择列表中的任一项。

（3）设置速度向量对话框中的其它选项。

（4）单击 Display 按钮在激活的窗口中绘制速度向量图。

2．速度向量绘制选项

在生成步骤中的第 2 步提到的其它选项包括向量箭头比例、颜色、间隔等，除了这些参数，还可以设定其他一些参数包括忽略一些向量的显示，使用相对速度梯度（为了移动参考面和移动网格），在数据表面显示速度向量，显示固定长度或固定颜色的向量，显示速度向量各方向的组成，指定显示的变化范围，对一个纯量场的速度向量进行色彩渲染，显示部分网格的速度向量，改变箭头的类型和比例等。

大部分常见选项都是通过速度向量对话框的选项进行设置，而一少部分则是在向量选项对话框中设置的（如图 12-3-3），可单击速度向量对话框中的向量选项（Vector Options）打开该对话框

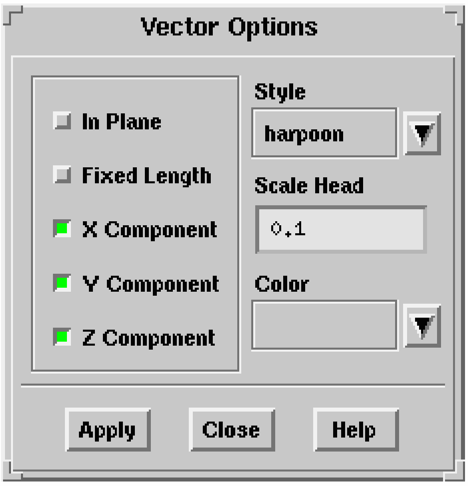


图 12-1-20 向量选项对话框

默认情况下，速度向量是自动缩放，以便在没有任何向量被忽略时重叠的向量箭头最少。可以修改自动比例（Auto Scale）的大小。采用自动比例的优点时当比例系数为 1 时，不管求解对象的尺寸，向量图显示将会很适合，给了一个好的调节比例系数的起点。

如果关闭自动比例（Auto Scale）选项，速度向量将会被按照实际的尺寸和比例系数（默认为 1）进行绘制。一个向量的“实际”尺寸时该点的速度梯度。一个速度梯度为 100m/s 的点其向量将被绘制成 100 米长，不过求解对象是 0.1m 还是 1000m。可以在速度向量对话框中通过修改向量比例系数的值使该向量尺寸（实际尺寸和比例系数的乘积）达到要求。

如果向量显示图上包含了太多的箭头导致图形显示混乱，可以通过设置速度向量对话框

中的跳过（Skip）值为该向量图“瘦身”。默认情况下，Skip 的值为 0，这意味着每个求解对象或平面上的向量都被显示（假设有 n 个向量）。如果将 Skip 增大到 1，将会有 n/2 各个向量。如果增大到 2，将会产生 n/3 个向量，等等。对表面的选择（或求解对象单元）将会决定哪一个向量被忽略或被绘制；因此当 Skip 的值不为 0 时，调整选择顺序将会改变速度向量图。

如果在求解问题时使用了一个或多个移动参考面或者移动网格，那么可以选择显示绝对速度向量或者相对速度向量。选中速度选项对话框中的 Relative 选项时，会按照设定的参考值以参考面为基础绘制向量。参看设置参考区来获得更详细的细节（如果建立了一个旋转参考面，不需要指定参考区域，会以旋转参考面为基础绘制速度向量）。如果选中相对（Relative）选项，在绘制向量图时会以绝对、固定的参考面为基础进行绘制。

对于一些可能对垂直于流场部分的可视化感兴趣的问题，这些“二次流”部分通常比沿着流动方向的部分面积要小，因此当流动方向部也显示时候就很难观察该方向流动情况。为了方便的观察垂直流场部分，可以在向量选项中选中平面内（In Plane） 选项。当该选项被选中时，FLUENT 只显示选中面内的速度向量图。如果选中的表面时一个交叉的流对象，将会显示垂直于该流场的速度向量图。图 12-3-4 显示了选中平面内（In Plane）时生成的速度向量图（注意这些向量被转化到求解对象的外部，正如在一个场景中改变求解对象的外形中所述，所以可以很方便的被观察）

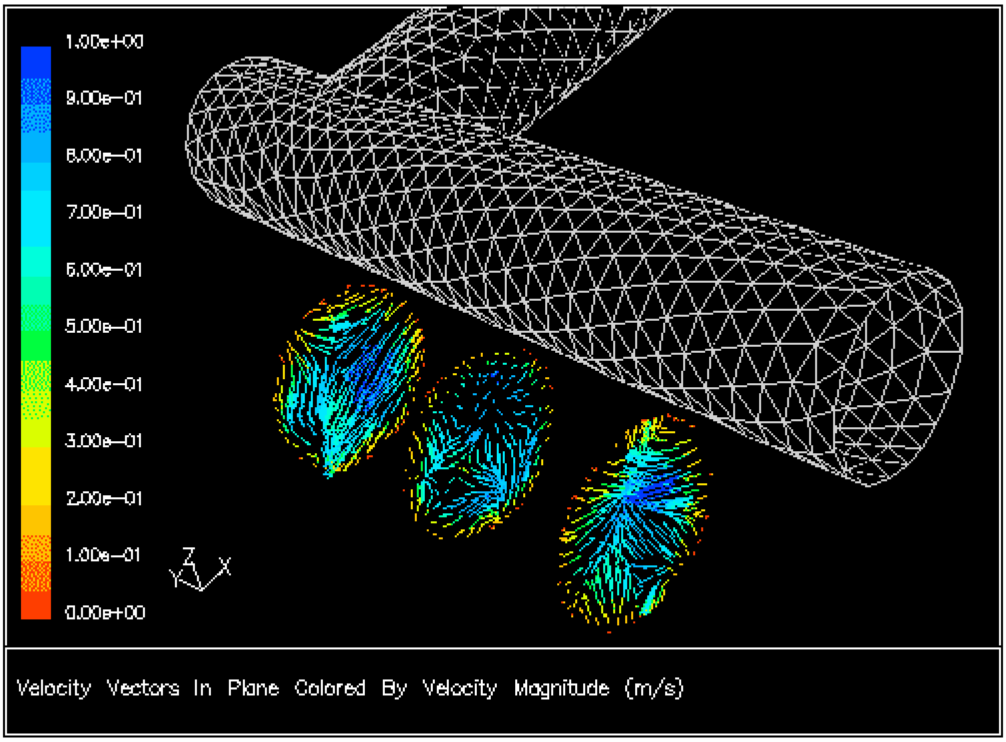


图 12-1-21: 使用平面内（In Plane）选项生成速度向量图。

在默认情况下，一个向量的长度和它的速度梯度成正比。如果希望所有的向量以相同的长度进行显示，则可以通过在向量选项对话框中选中固定长度（Fixed Length）选项实现。要修改向量长度，可以在速度向量对话框中调整比例系数的值。

默认情况下，将显示速度向量的各个笛卡儿坐标的分量，所以箭头指向为沿着物理空间的矢量方向，有时一个方向上的分量非常大，例如 x 方向。在这种情况下，可能希望缩小 x

方向的分量以便观察 y、z 方向的份量。要减小一个或多个速度向量分量，请在向量选项对话框中关闭相关选项（X, Y 或 Z 分量）。

默认情况下，在速度向量显示中包含的速度变化范围通常时按照求解对象的速度梯度变化范围进行设定的。如果想观察一个小范围内值的变化，可以重新限定显示的范围。用来代表速度向量显示的色彩将会随之显示范围值的变化而变化。（也可以按照速度梯度颜色的渲染将一个纯量场改变为其它纯量场）。

手工设置速度梯度的变化范围（或者通过颜色下拉列表框改变纯量场的范围），可在速度向量对话框中关闭 Auto Range 选项。Min 和 Max 编辑框处于可编辑状态，然后可以在其中输入新的范围值。例如，需要显示速度梯度大于 150m/s 而且小于 300m/s 的区域，可以将 Min 设置为 150 而 Max 为 300。同样，如果对静态压力向量进行渲染，可以只显示压力为某一个特定值的曲艺。在显示默认范围的时候，单击 Compute 按钮将更新最小及最大值。

当限制了速度向量显示的范围后，也可以控制超出这个范围的值是否显示。当选中 Clip to Range 选项时（默认选项），不显示超出设定范围的值。而当该选项未被选中时，高于最大值的值将以代表最大值的颜色进行显示。这一点和显示色彩填充等值线相似。（参看等值线轮廓绘制选项中的图 2、图 3）

如果希望以其它量场而非速度量场对向量进行渲染，可以通过在 Color By 下拉列表框中选择一个不同的变量或函数来实现。按照上面的列表选择所希望的分类，然后从下面的列表框中选择相关量。如果选择了静态压力，例如，速度向量将仍和速度梯度有关，但是速度向量的颜色将和每一点的压力有关。

如果希望所有的向量都以相同的颜色进行显示，则可在速度选项对话框中的 Color 下拉列表框中指定所使用的颜色。如果没有选择任何颜色（例如，下拉列表框中最上面的选项也是默认选项——空格），向量的颜色将由速度向量对话框中的值为 field 的 Color By 的选项决定。单色向量显示通常在等值线和速度向量叠加图中很有用。

对于一些问题，尤其是复杂的 3D 几何体，很可能希望在的向量图中包含部分网格作为空间参考点。例如，想在速度向量图中显示入口和出口的位置。上述任务可以通过在速度向量对话框中打开 Draw Grid 选项来完成。当选中 Draw Grid 选项时，网格显示对话框将自动打开，可以通过该对话框设置网格显示参数。当单击速度向量对话框上的 Display 按钮后，按照在网格显示对话框中的设置，在速度向量图中将会显示部分网格。

Fluent 总共有 5 中不同类型的箭头提供给使用。可在 Style 下拉列表框中选择 cone，filled-arrow, arrow, harpoon 和 headless 类型。默认箭头为 harpoon 类型。

如果选则的箭头类型中包含了头部，可以通过向量对话框中的 Scale Head 来修改箭头的大小尺寸。

12.1.4 显示轨迹

轨迹被用来显示求解对象的质量微粒流。粒子由在 Surface 菜单中定义的一个或多个表面中释放出来（参看为显示图形和数据产生表面）。现形或楔行面经常被使用（参考线形和楔形面）。图 12-4-1 显示了一个轨迹图。

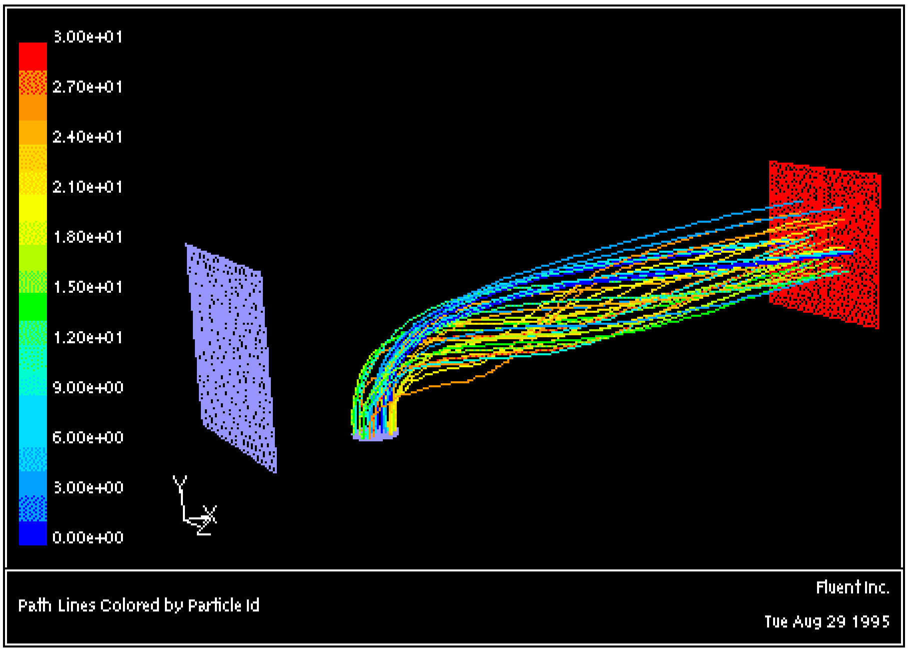


图 12-1-22：轨迹图

1．产生轨迹的步骤

（1）可以通过轨迹线对话框来完成轨迹的生成 (如图 12-4-2).

菜单：Display –〉Path Lines...

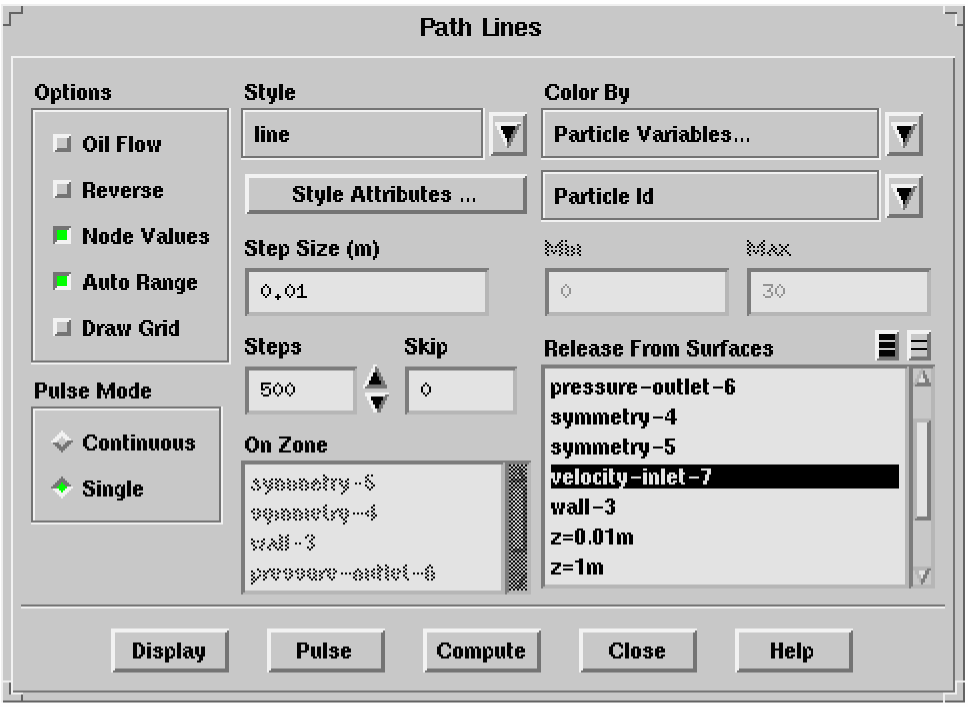


图 12-1-23：轨迹线对话框

（2）在 Release From Surfaces 列表中选择相关平面。

（3）设置 Step size 和 Steps 的最大数目。Step Size 设置长度间隔用来计算下个各微粒的位置。(注意当一个微粒进入/离开一个表面是其位置通常由计算得到；即便指定了一个很大的

Step Size，微粒在每个单元入口/出口的位置仍然被计算并被显示)。Steps 设置了一个微粒能够前进的最大步数。当一个微粒离开求解对象并且其飞行的步数超过该值时将停止。如果希望微粒能够前进的距离超过一个长度大于 L 的求解对象，一个最简单的定义上述两个参数的方法是是 Step Size 和 Steps 的乘积应该近似等于 L。

（4）设置轨迹线对话框中的其它选项。

（5）击 Display 按钮绘制轨迹线，或者单击 Pulse 按钮来显示微粒位置的动画。在动画显示中 Pulse 按钮将变成 Stop 按钮，可以通过单击该按钮来停止动画的运行。

12.1.5 显示扫过面（sweep surface）结果

扫过面用来检查计算域的网格，等值线，向量等而无须创建相应的面。

1．生成扫过面步骤

（1） 打开扫过面面板。

菜单：display->surface

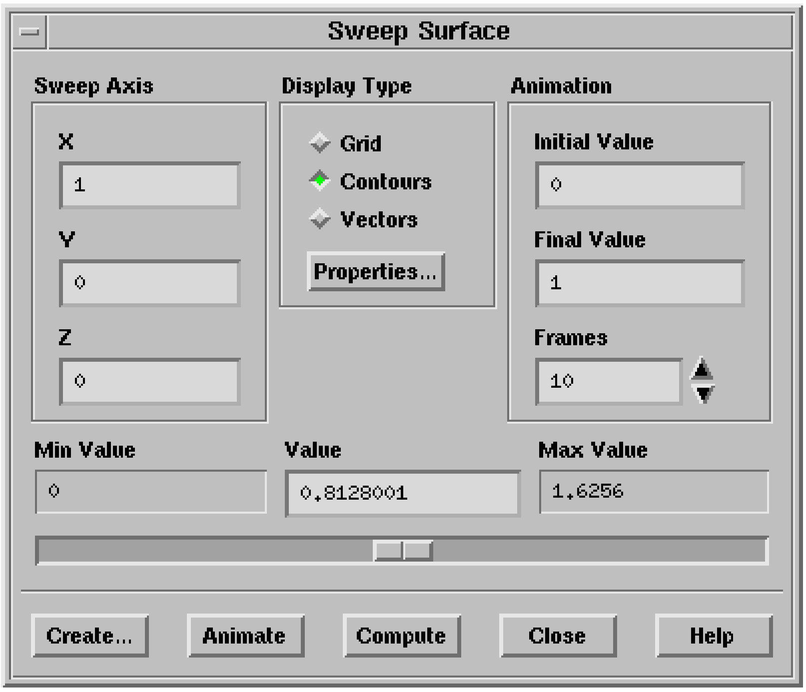


图 12-1-24 扫过面面板

(2) 指定扫过面的基准轴，用来选择方向（x,y,z)。

（3）点击计算（**Compute** ）更新最大最小值。

（4）在显示方式中（display type）指定显示方式，网格、等值线或者向量。

（5）拖动滑动条，观察数据。

(6) 如果希望保存现有面数据，可以点击生成（creat）按钮，生成面。

2．动态扫过面显示

观察动态扫过面的步骤如下

（1）指定轴。
（2）在动画（animation）选项中，输入要动态显示的起始值（initial value）和结束值（final value）。
（3）指定动画祯数。
（4）点击动态（animate）按钮。

12.2 调整图形显示方式

生成基本图形后，有许多种方法来调整图形的显示。经常用到的包括如下几种：

1. 图形的重叠显示
2. 输入标题和注释
3. 改变图形渲染方式

12.2.1 图形的重叠显示

图形的重叠显示可以用来在一幅图表中显示不同的基本图表，可以把等值线图和矢量图一起显示，这样更容易观察数据的变化。图形的重叠显示步骤非常简单，只要打开重叠显示功能，在场景菜单中选择重叠就可以了。

菜单：display->scene

一旦激活了重叠选项，后面所生成的所有图形都会在图形窗口中重叠显示，要想生成不重叠的图形必须首先确保没有选中重叠选项。

重叠显示时，图标的标题和色彩范围为最后生成图表的标题和色彩范围。图 12-2-1 为等值线图和矢量图重叠显示的例子。图 12-2-1 等值线图与矢量图的重叠显示

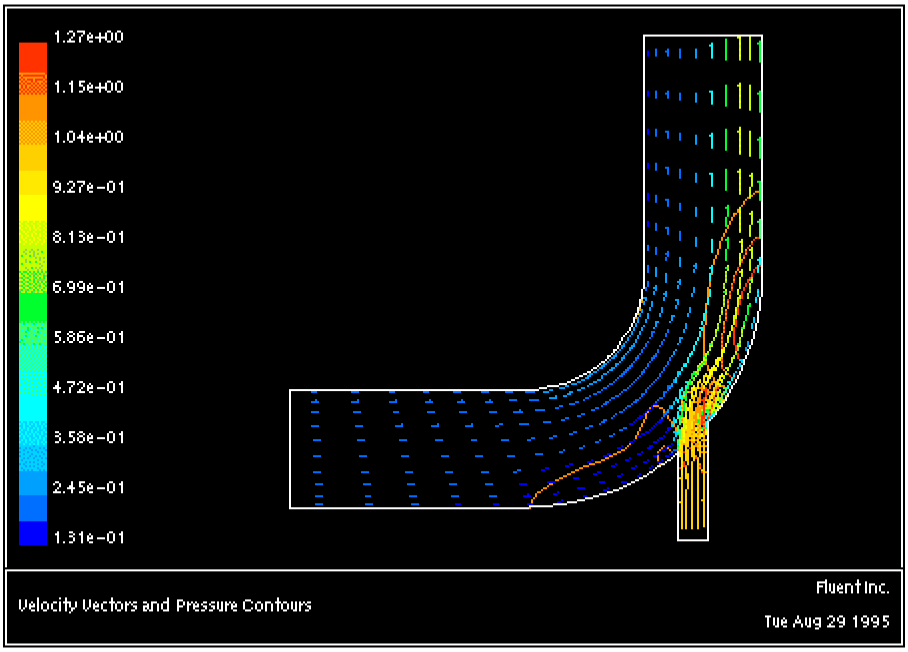


图 12-2-1 等值线图与矢量图的重叠显示

12.2.2 输入标题和注释

Fluent 生成的图形在默认情况下都包含有标题，用来显示计算的主要数据，例如时间，模型等等。如果希望改变标题的内容、大小、颜色等，可以用用文本命令或者显示面板来完成。

1. 使用/不使用标题和色彩范围

可以选择不显示标题，只要在显示面板选项中确定标题项不被选择就可以了。

菜单：display->options

注意可以用文本命令进行标题的修改。命令为：

Display->set-〉windows->text 修改文本文字

Display->set->windows->scale->修改大小

2. 修改标题

添加标题

输入命令 display->set->title，就可以输入标题了。注意必须用双引号（“”）把标题包起来，否则会提示无效。

3. 使用/不使用轴线

只要在显示选项面板中不选择该项就可以了。

菜单：display->options

4. 添加注释

有两种添加注释的方法，可以用鼠标注释方法和用注释面板。图 12-2-2 为一个添加注释后图形的示例。

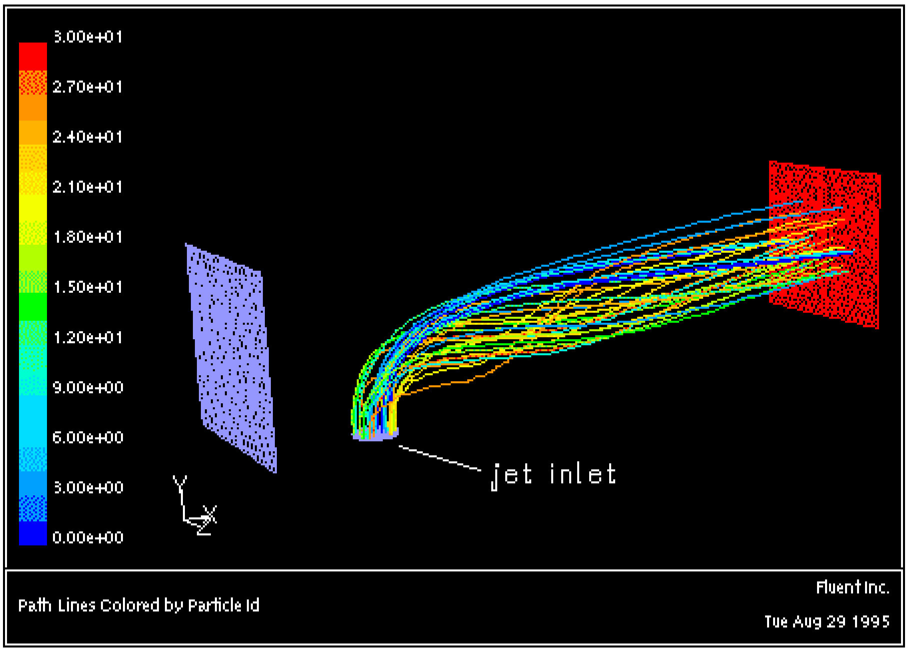


图 12-2-2：添加注释后的图形

（1）用注释面板添加注释

菜单：display->annotate

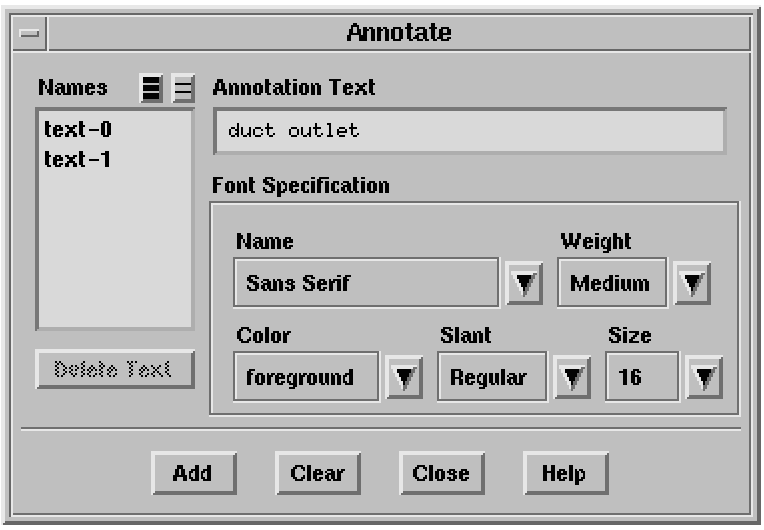


图 12-2-3 注释面板

1. 选择字体，大小，颜色。
2. 输入文字
3. 点击添加。

（2）用鼠标注释功能添加

首先需要在鼠标按钮面板中设定一个按键为注释键。然后用定义的注释键在图形中点击希望注释的位置，会出现一个光标便可以输入注释了。

（3）编辑存在注释

一旦添加注释，便可以用注释面板或者鼠标注释功能修改注释的字体，颜色等。

12.2.3 修改图形渲染方式

根据所计算的流场几何形状和图形硬件和软件的不同，可以修改图形渲染方式。菜单：display->options，打开显示选项面板。

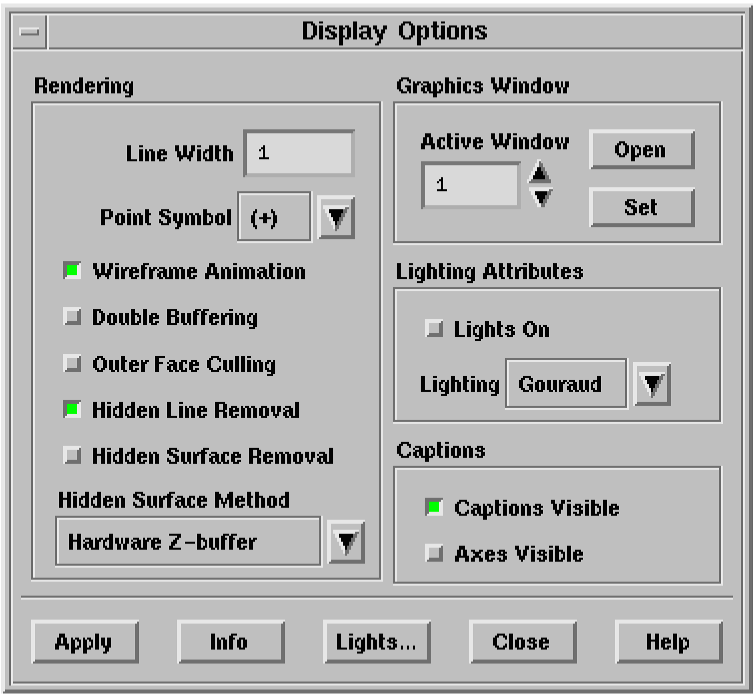


图 12-3-1 显示选项面板

注意在修改任何参数后，都需要单击应用按钮（apply）。

1．可以调整的参数包括：

（1）线宽，默认为 1 个像素。
（2）点符号：默认情况下用“+”号代表点，可以在下拉菜单中选择不同的样式。
（3）动态线框：当用鼠标调整显示时，会出现一个动态线框。

（4）双倍缓存：可以极大地减少显示器的不平稳。
（5）外部面显示选择：可以选择不显示外壁面。
（6）移动时隐藏线：推荐在 3D 时选择该项。
2．显示设备信息

可以点击显示选项中的信息(info)选项查看显示设备驱动程序和计算机适配器信息。

12.3 控制鼠标功能

在 Fluent 中可以方便的调整鼠标键的功能，根据不同的配置，在图线窗口中前即鼠标按键可以完成不同的工作。

1． 鼠标功能简介

（1） 鼠标旋转功能：水平拖动按键可以使图形沿着显示器的 y 轴旋转，而垂直拖动则可沿着 x 轴旋转。
（2） 鼠标移动：可以在按下按键后拖动图形。
（3） 鼠标放大：可以在按下按键后划出一个放大缩小窗口，如果左右划出窗口，当释放按键后则会放大，若右左拖动，则会缩小。
（4） 鼠标旋转放大：可以在旋转的同时放大或者缩小视图。
（5） 鼠标求值：在图形中点击按钮，在 fluent 文本窗口（即主窗口中）显示该图形信息。
（6） 鼠标注释：如 11 章鼠标注释所述，可以应用该功能方便的在图形窗口中添加注释。

2．调整鼠标功能按键

调整鼠标按键功能可以在鼠标按键面板中完成（mouse button panel）如图 12-3-1。

菜单：display->mouse buttons…

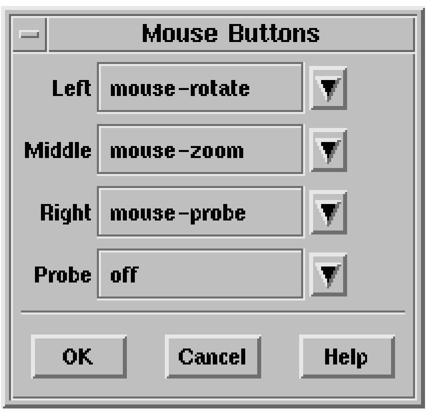


图 12-3-1 鼠标按键面板

对于鼠标的每一个按键，可以指定不同的功能。Fluent 软件默认的功能如表 12-3-1。

表 12-3-1 默认鼠标功能

按键	2D	3D
左	拖动	旋转
中	放大	放大
右	求值	求值

12.4 修改观察方式

可以利用 Fluent 软件提供的强大图形功能方便的调整图形窗口的观察方式，例如移动，放大等等功能。也可以保存视图为图像，或者进行删除，恢复等等操作。这些操作都可以视图面板中完成。（图 12-4-1）。

12.4.1 放大、居中、旋转、移动以及放大缩小功能

这些操作都可以在视图面板中完成。如图 12-4-1。

菜单：display->views…

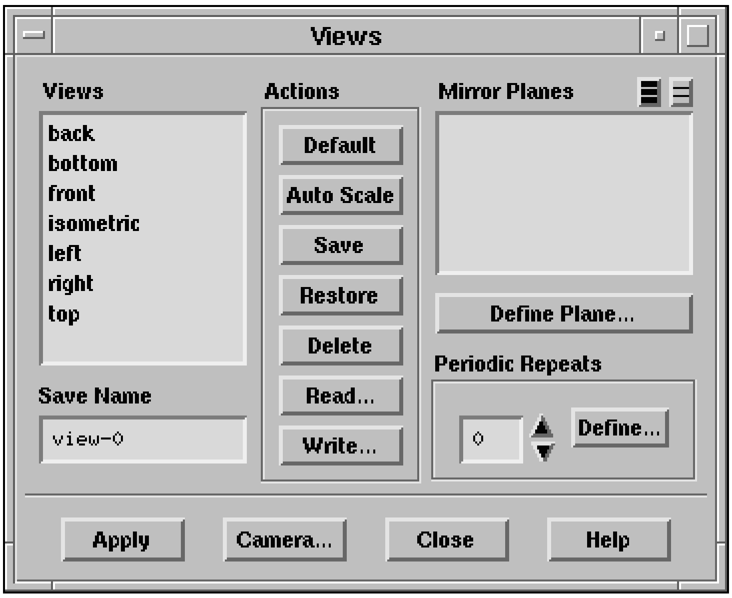


图 12-4-1：视图面板

可以用鼠标或者照相参数面板来完成图像的旋转放大、缩小、移动功能。照相参数面板如图 12-4-2。

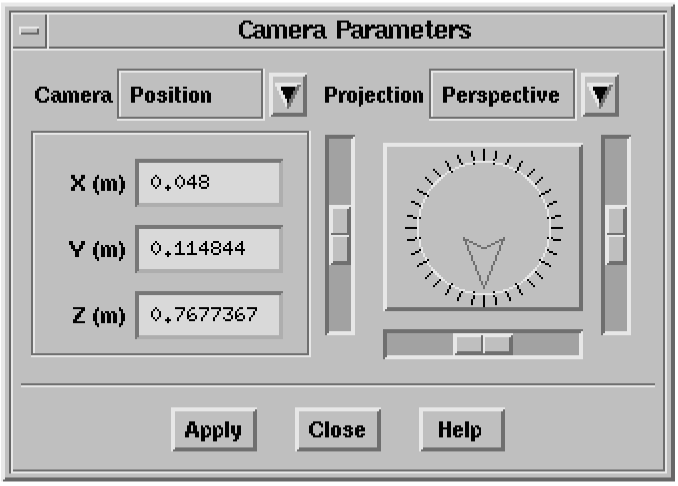


图 12-4-2：照相参数面板

12.4.2 控制透视图和照相参数

透视和其他的照相参数在照相参数面板中进行设置。

1. 透视图和全视图

可以选择显示透视图或者类似于机械制图中的全视图图形，该选项位于照相参数面板中的投影（projection）项。

2. 修改照相参数

除了前面提到的移动、旋转、放大/缩小视图，也可以用调整“照相机”参数的方法改变视图。

照相机共有四个参数：位置、目标、上向量和场。四个参数的关系如图 12-4-2。

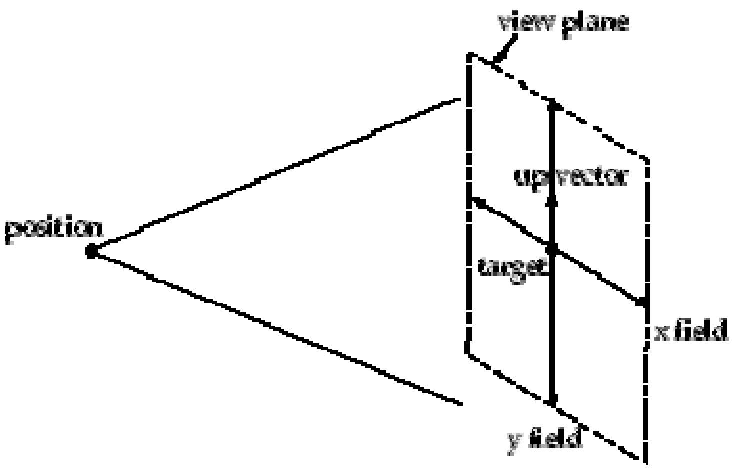


图 12-4-2：照相机定义

12.5 构建场景

当生成了某种视图以后，往往需要调整其参数来增强图形的效果。这些调整可以在视图描述面板中完成（scene description panel）。图 12-5-1。

菜单：display->scene

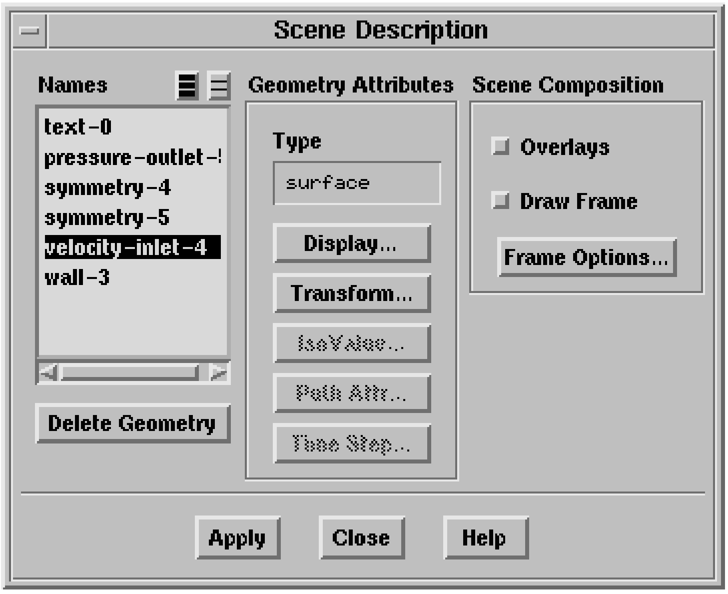


图 12-5-1：视图描述面板

12.5.1 选择操作对象

为了使图中的对象进行操作，必须首先选择对象，使其成为当前的工作对象。对象的选择非常简单，只要在名称列表中选择就可以了。可以选择一个或者多个对象同时进行操作。

12.5.2 改变对象显示方式

为了增强图形的显示效果，可以改变对象的颜色、可视性以及视图中其他对象的参数来达到目的。例如，如果想显示一个复杂问题的全部网格，可以选择隐藏部分网格而达到突出边界等部分网格的目的。这些功能都可以在显示属性面板中完成（display properties panel）。图 12-5-2。

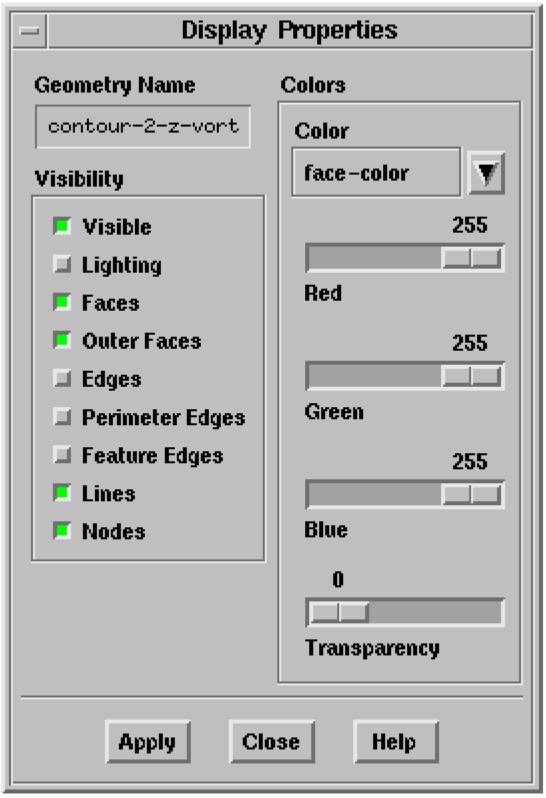


图 12-5-2：显示属性面板

12.5.4 移动场景中的对象

在构建场景时，往往需要把某些对象从初始位置移动以突出显示效果。如图 12-5-3 就是一个把三个位置的矢量图移开的示例。要完成这种操作也非常简单，只要选择好移动对象，然后在移动面板中输入合适的位移距离，或者旋转等其他参数就可以了。位移面板如图 12-5-4 所示。

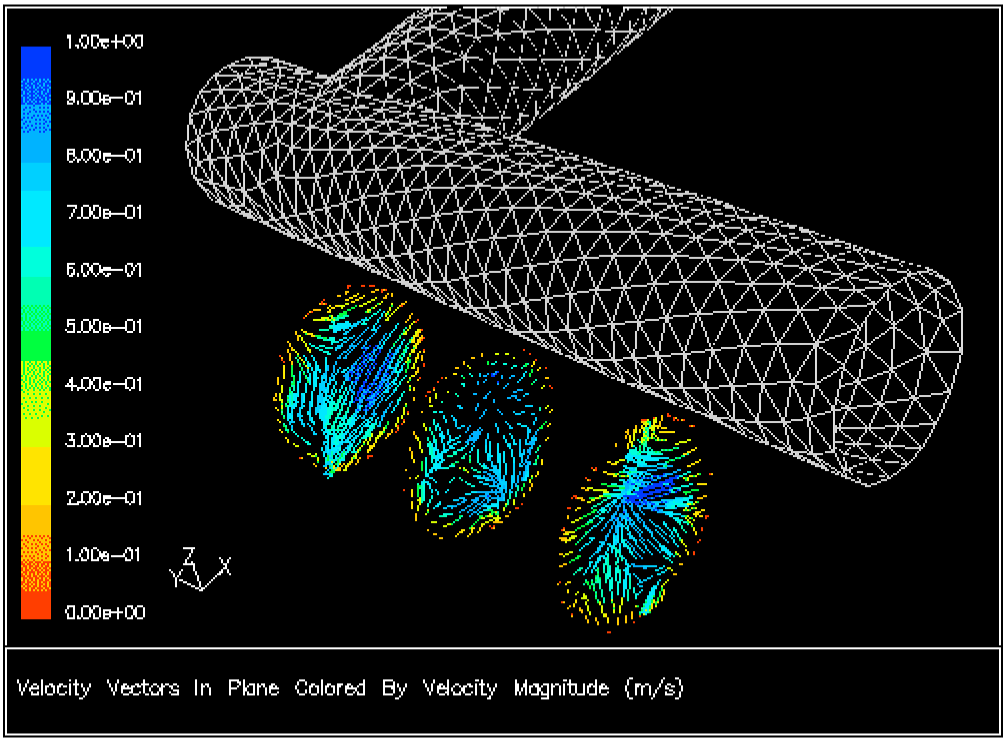


图 12-5-3：速度向量图移动示例

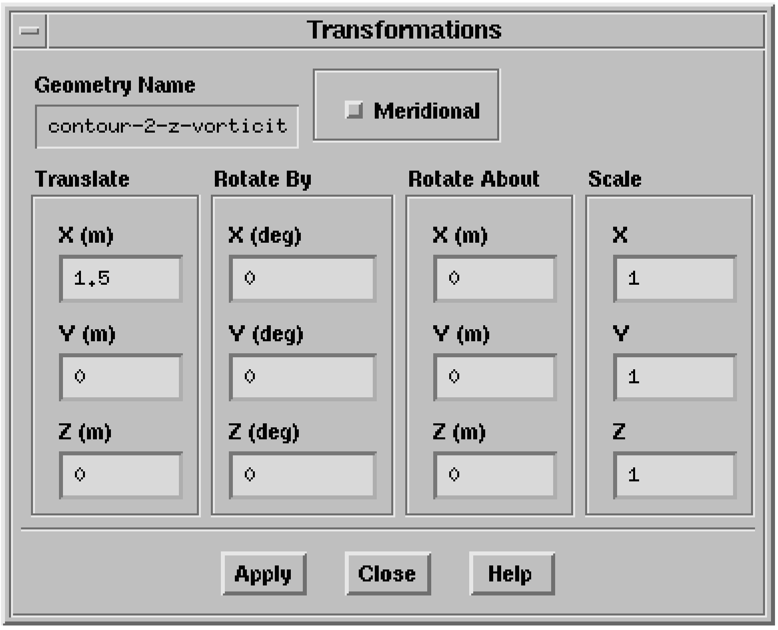


图 12-5-4：移动面板

12.6 动画技术

在 Fluent 软件中可以生成关键祯动画，通过把静态的图像转化为动态的图像，可以大大加强结果的演示效果。动画的创建需要在动画面板中完成。图 12-6-1。

菜单：display->scene animation

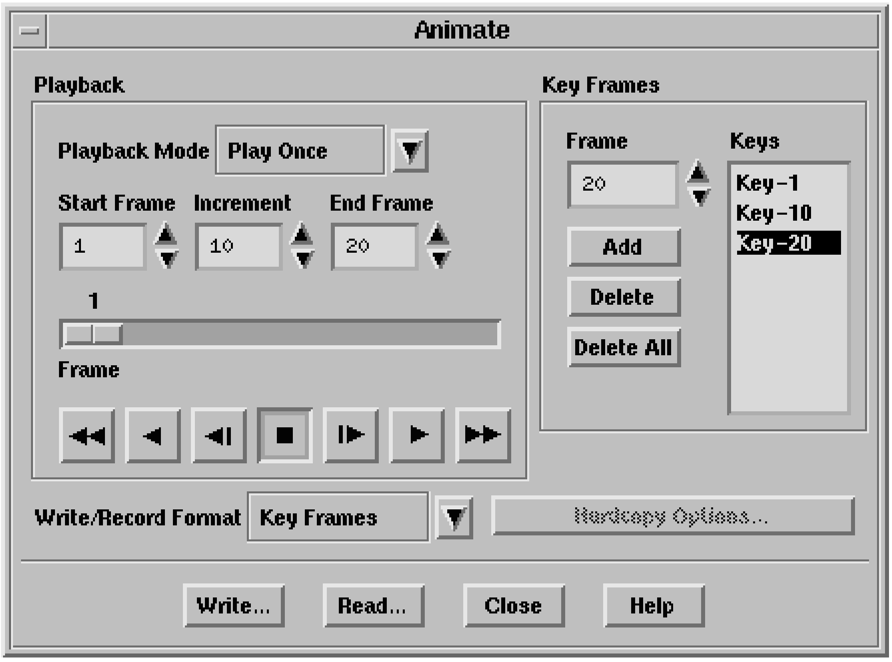


图 12-6-1：动画面板

另外利用求解动画面板（solutions animation panel）还可以创建时间步动画。

12.6.1 动画的创建

创建关键祯动画的步骤如下：（最多 3000 祯）。

1. 输入祯数。
2. 选则需要的关键祯。可以包括不同视角的观察。
3. 选择关键祯的时间。
4. 利用关键祯构成动画。
5. 可以回放检查效果，满意后选择保存。

12.6.2 动画的保存

动画创建完成后可以进行保存方便以后的查阅和结果的展示，Fluent 支持四种动画格式：

1. 动画文件（Fluent 专用）。
2. 硬拷贝文件，保存动画的每一祯。
3. MPEG 文件。
4. Video 文件。

1．动画文件

改种文件只可以被 Fluent 软件识别和读取数据，特点是文件小，不失真。

2．硬拷贝文件

改种文件把动画的每一祯生成一个图像，可以为不同的图像格式（例如 jpg,bmp 和 tiff 格式）。生成硬拷贝动画文件的步骤如下：

1. 在面板中的 write/record format 中选择硬拷贝（hard copy）.
2. 在硬拷贝选项中设定图像参数。
3. 在动画面板中点击写（write）开始保存文件。

12.6.3 读取动画文件

动画文件的读取非常方便，首先单击动画面板中的读取（read）按钮，会打开选择文件面板（selece file dialog panel）,只要选择目标文件便可以打开动画文件。

12.7 柱状图与 XY 插值曲线

除了上面提到的图形工具，Fluent 还提供了 XY 曲线工具，可以方便的柱状图与数据结果绘制 XY 曲线，可以修开曲线的颜色，线形，比例尺，轴线，曲线属性等等，来提高 XY 曲线的表达效果。

12.7.1 绘图类型

1．XY 曲线图

XY 曲线由线或者数据点构成。理论上所有的变量可以用该功能绘制曲线。另外，甚至可以读入由其他软件生成的数据文件来比较计算结果。图 12-7-1 为一个 XY 曲线示例。

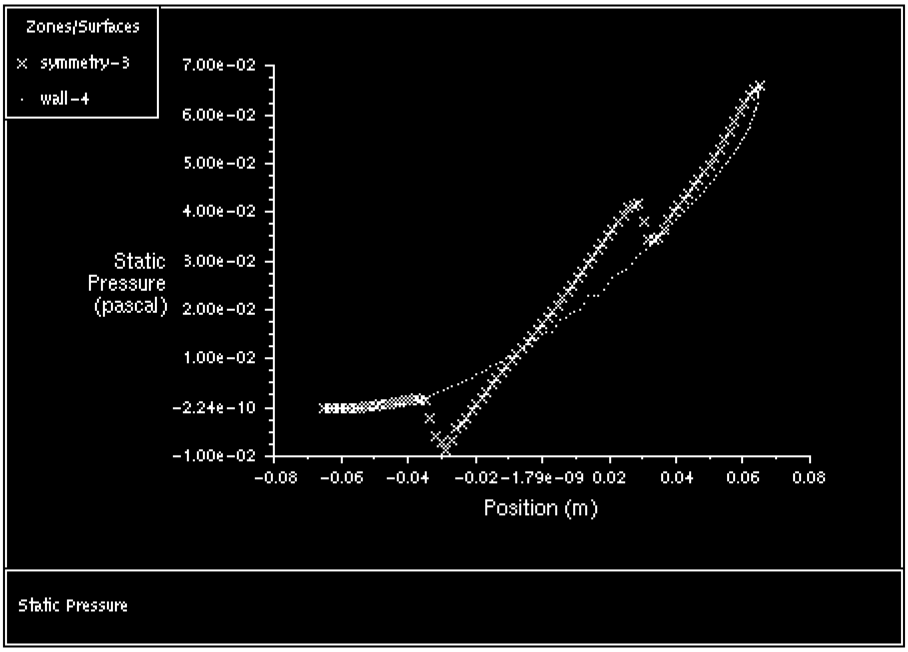


图 12-7-1：XY 曲线示例

2．柱状图

柱状图为柱状图，由柱状图的高低代表数据的频率。可以对柱状图添加注释。图 12-7-2 为一个柱状图示例。

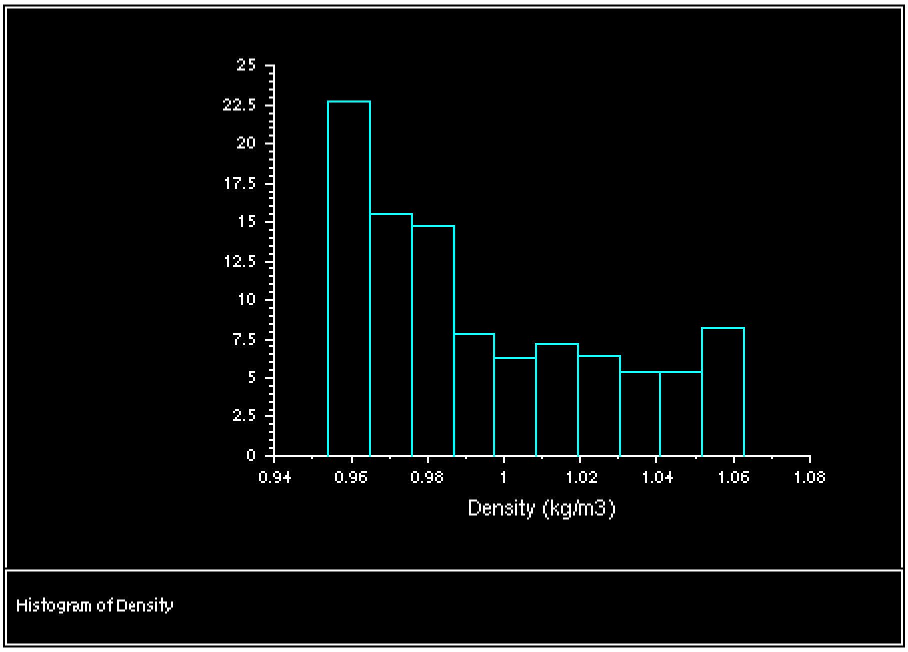


图 12-7-2：柱状图示例

12.7.2 计算数据 XY 曲线绘制

1．生成 XY 曲线步骤

(1)打开计算结果 xY 曲线面板（所露体 XY 曲线面板）。如图 12-7-3。

菜单：plot->XY plot

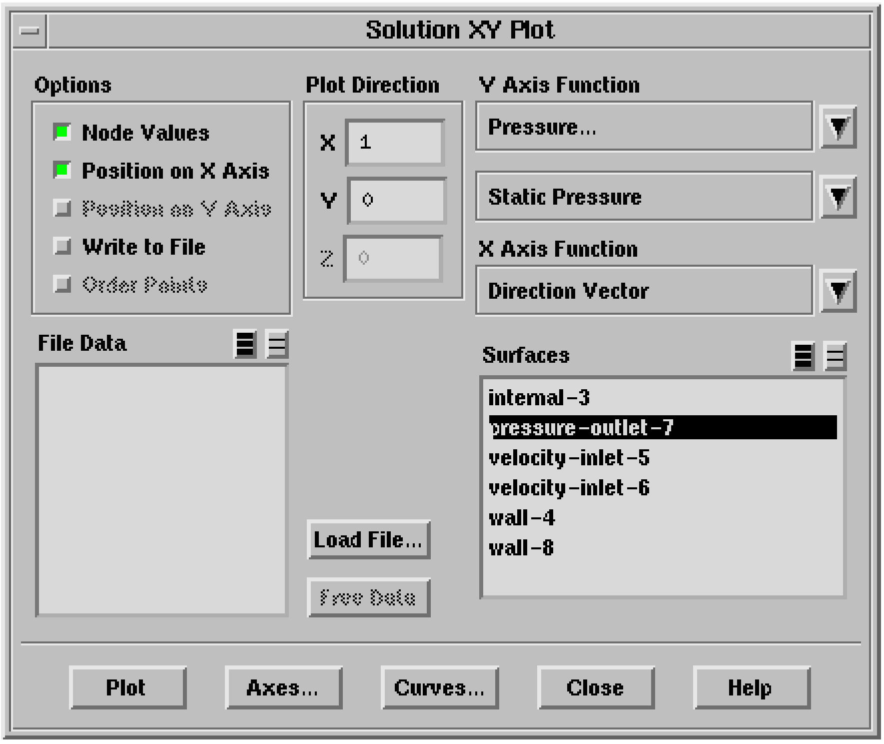


图 12-7-3 计算结果 XY 曲线面板

（2）选择绘图变量。

（3）选择绘图方向

（4）选择绘制曲线的表面。

（5）单击 plot 按钮，绘制曲线。

可以利用鼠标按钮方便的添加注释。如图 12-7-4，12-7-5，12-7-6 所示。

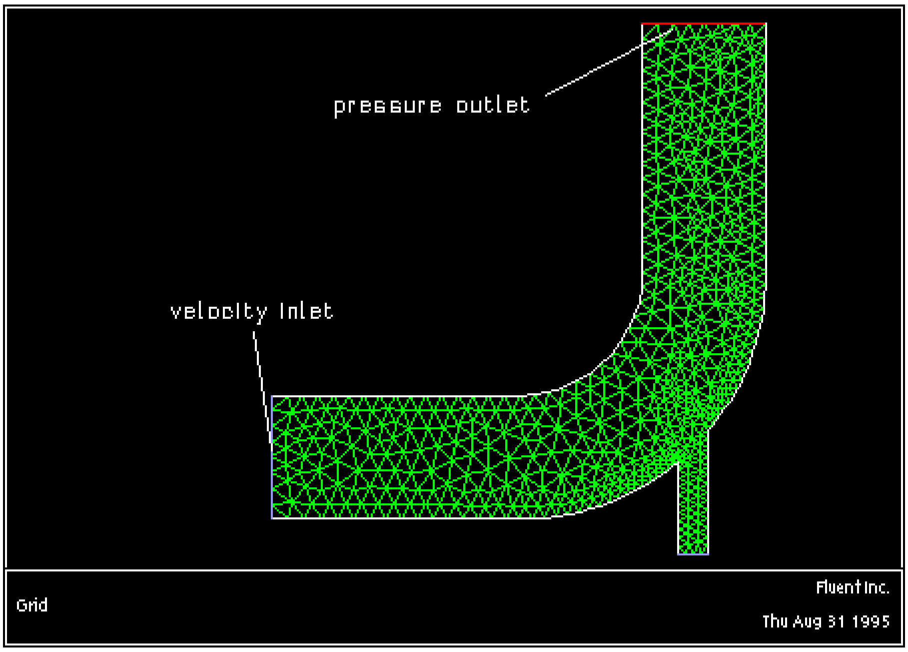


图 12-7-4：用来绘制 XY 曲线的几何体

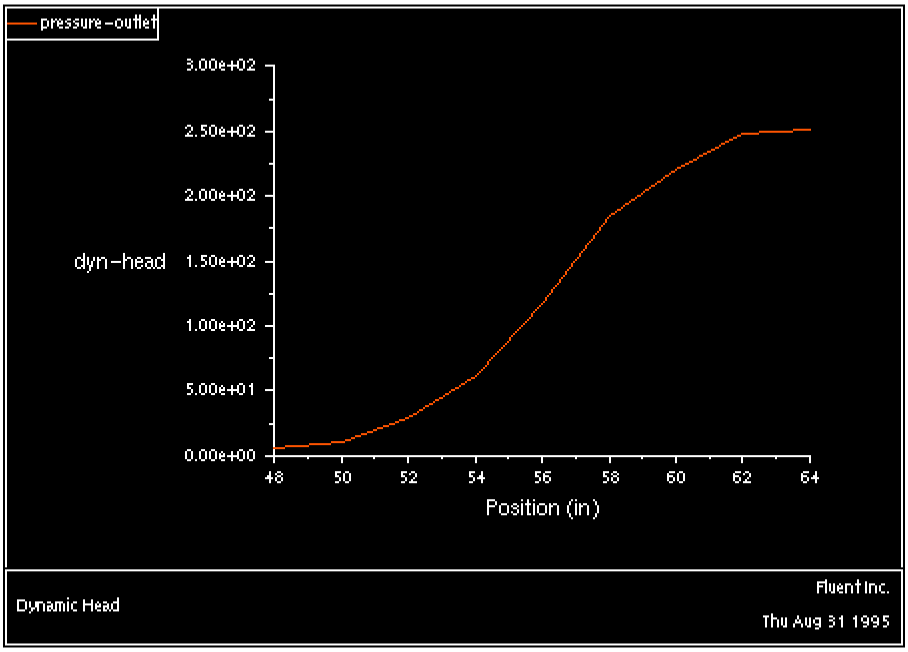


图 12-7-5：出口 XY 曲线，绘图方向（1，0，0）

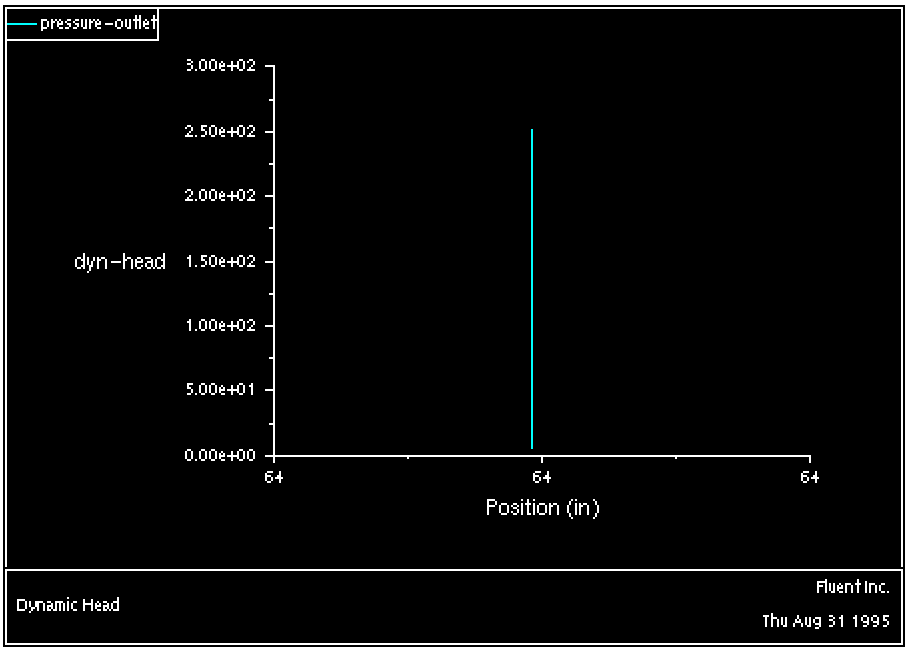


图 12-7-6：出口 XY 曲线，绘图方向（0，1，0）。

12.7.3 XY 曲线格式

由 Fluent 软件生成的 XY 曲线文件包含如下信息：

1. 曲线名称、
2. 坐标系名称
3. 曲线变量与成对数据点

下列格式为 XY 文件的示例：

```
 (title "Velocity Magnitude")
(labels "Position" "Velocity Magnitude")

((xy/key/label "pressure-inlet-8")
 (xy/key/visible? #t)
 (xy/line/pattern "--")
0.0000 230.097
0.0625 160.551
0.1250 149.205
...
0.5000 183.007
)
```

类似于算例文件格式，括号内为各种信息，类似于一般 ACSII 文件。

12.7.4 计算结果柱状图

柱状图可以在图形窗口中显示或者在对话窗口中显示。柱状图的横坐标为变量的值，纵坐标为是占总网格数的百分比。

1. 生成柱状图步骤

（1）打开计算结果柱状图面板。如图 12-7-7。

菜单：plot->histogram…

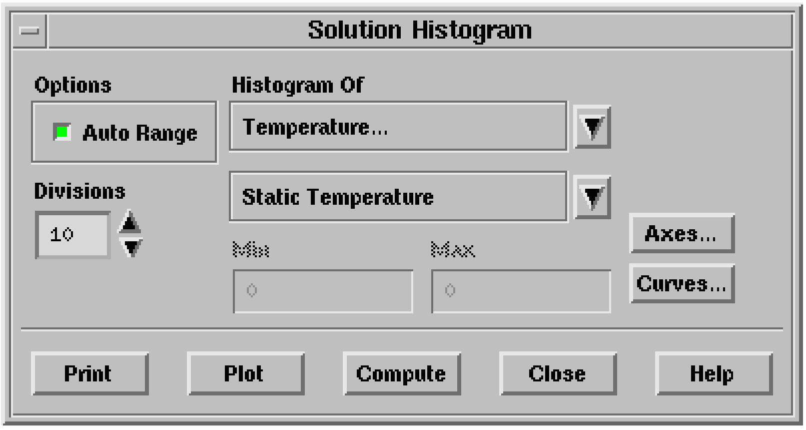


图 12-7-7： 求解柱状图面板

（2）选择标量。
（3）设定数据间隔点，默认情况下为十个间隔。
（4）设定曲线格式。
（5）单击 plot 按钮，生成图表。

2. 柱状图参数

除了前面提到的轴线和曲线属性，唯一还可以进行调整的参数便是柱状图的次范围。

（1）为柱状图设定范围

在默认情况下，柱状图的绘制范围为全部的变量值，如果希望只绘制较小的范围，可以手工限定范围。

要想手工设定范围，首先要首先除掉面板中的自动范围(auto range)。然后就可以编辑范围的最大最小值，要显示默认的范围，只要点积极宋（compute）按钮就可以了。

12.7.5 修改坐标轴属性

可以通过修改 XY 曲线和柱状图的坐标轴格式来调整图表的标题，尺寸，范围，数目，主要尺寸和最小标尺。对于每一种曲线，XY 曲线，残差曲线，和柱状图，都可以在坐标轴（axes panel）面板中设置坐标轴格式，如图 12-7-8。

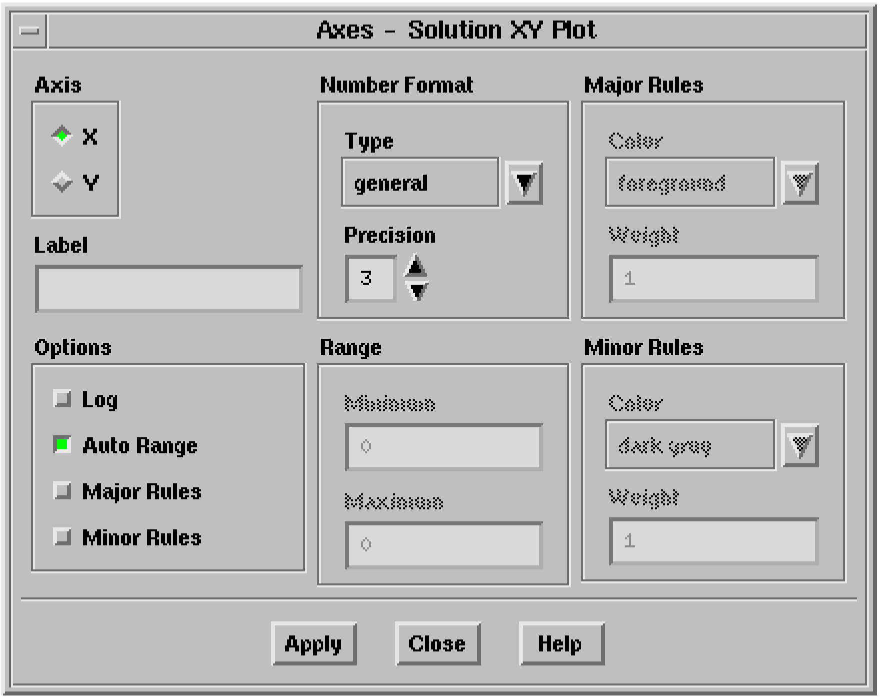


图 12-7-8：坐标轴面板

要打开坐标轴面板，可以单击绘图窗口中的坐标轴（axes）按钮就可以了。

12.7.8 修改曲线特性

XY 曲线与直方图中的曲线可以综合。使用任何 Fluent 软件提供的线形和符号。可以用曲线面板（cureve panel）修改曲线的样式、宽度、颜色、符号、尺寸、符号颜色等属性。面板如 12-7-9。

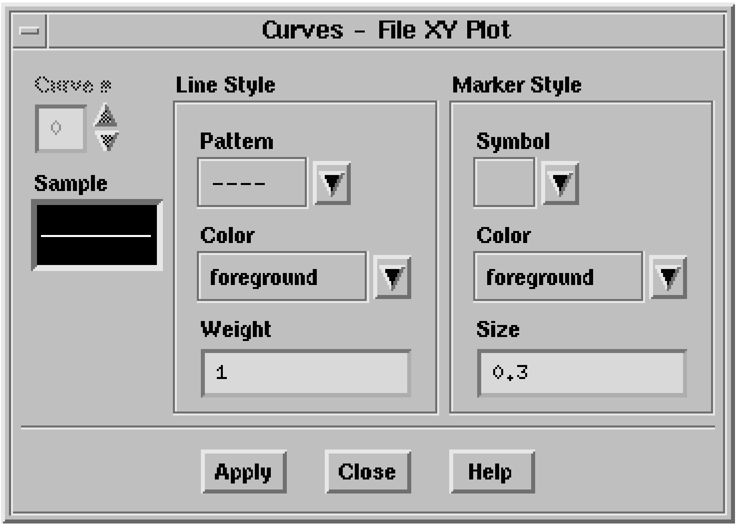


图 12-7-9：曲线面板

由于曲线属性的设置与 Excel 电子表格中的设置基本一样，这里不一一介绍其用法。打开曲线面板的程序为在各种绘图面板

第十三章 计算报告

在后处理过程中，可以利用 FLUENT 提供的工具计算边界上或内部面上各种变量的积分值，可以计算的项目包括边界上的质量流量、热量流量、边界上的作用力和力矩、流场变量的平均值和质量平均值，并可以设置无量纲系数的参考量、计算几何体的投影面积、绘制几何数据和计算数据的柱状图，最后可以打印或者以文件形式保存一个包括模型参数、边界条件和求解参数设置等信息在内的简要报告。

在学习本章时须注意：对于二维平面问题，积分值为变量在单位厚度（垂直于屏幕方向）上的积分值；对于轴对称问题，积分值为整个旋成体内的积分值。

13．1 边界通量的计算

在所选择的边界区域上可以计算三种不同变量的流量，即质量流量、总的热流量和辐射传热流量。三者的求解方式分别如下：

1．边界上的质量流量，由边界区域各个网格面上的质量流量相加得到，各个面的质量流量等于密度乘以法向速度再乘以所在平面的投影面积。

2．边界上总的热流量，由各个面的热流量相加得到。各个面的热流量等于对流传热与辐射换热之和。热流量的计算与计算中设置的边界条件有关。例如，在一个温度不变的墙面上，热传导的热量等于热传导率乘以投影面积再乘以温度梯度（傅立叶定律）。根据模型的不同，总的热流量可能包括对流换热、焓的对流通量和能量的扩散通量等。

3．边界上的辐射换热量，等于每个面上的辐射换热量之和。辐射换热量的计算取决于计算中所使用的辐射模型，详情请参考与辐射模型相关的内容。

掌握了这三种计算方式，在求解边界流量值时便不会犯错误，也不会拿到了数据却不会分析，不知道怎样对数据进行二次加工。下面举一个很简单的例子说明，对于一个封闭系统如何得到它的质量流量？如何判断质量是否守恒？很简单，把进口和出口边界流量相加便是系统质量与零值的差异，若该数值非常小。我们便可以认定计算质量守恒。然后就可以用进口的流量或者出口的流量作为系统的流量数值。

对于包含入射粒子或者质量源的流量计算则与上面的算法稍有不同，主要的不同之处

在于，用上述方法得到的边界流量中不包括人为定义的入射粒子或者质量源的影响，因此可能出现质量或者能量不守恒的情况。那么怎样才能判定粒子和质量源存在时的计算是否满足质量或者能量守恒条件呢？很简单，可以把边界流量计算的数值与粒子轨迹报告中得到的质量和热量改变的数值相比较，如果两者相等，就可以认为满足守恒条件。

13.1.1 生成通量报告步骤

生成通量报告的步骤非常简单，只需要使用 Flux Reports（通量报告）面板便可以生成边界上的质量流量、热流量或者辐射换热量。流量报告面板（如图 13-1 所示）可以用下列菜单命令启动：

Report->fluxes

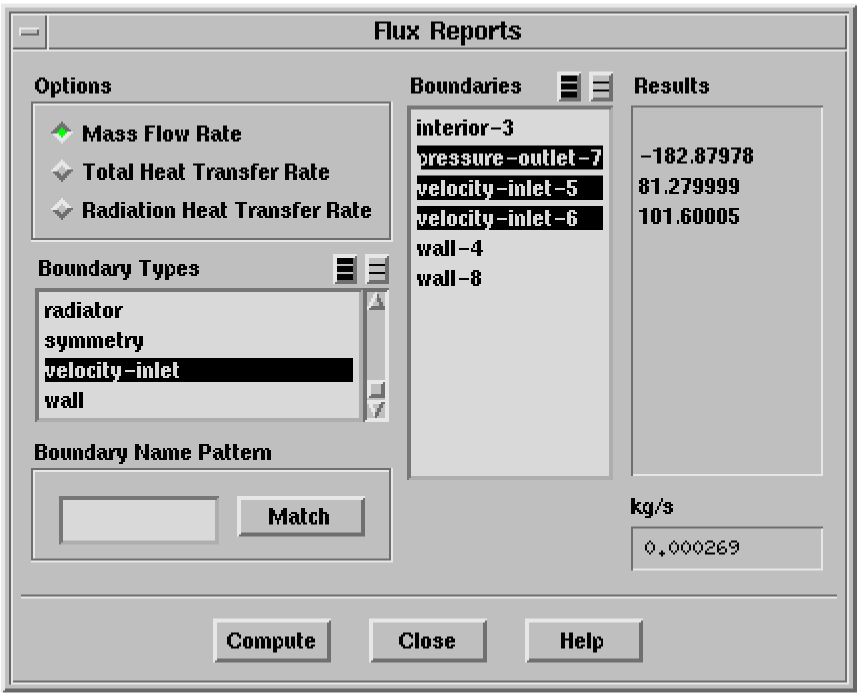


图 13-1 Flux Reports（通量报告）面板

生成报告步骤如下：

1．从 Options（选项）中选择计算变量：Mass Flow Rate（质量流量）、Total Heat Transfer Rate（总的热流量）或 Radiation Heat Transfer Rate（辐射换热量）。

2．在 Boundaries（边界）列表中选择目标边界。

如果想生成所有相同类型的边界区域的流量报告，可以在 Boundary Types（边界类型）中选择某个类型来集中选择某类型下的所有边界。这样，所有与选定的类型相同的边界区域将在边界（Boundaries）列表中自动被选择。

另外一个选择边界的快捷办法是用选择 Boundary Name Pattern（边界名称样式）完成选择，首先在 Boundary Name Pattern（边界名称样式）的输入栏中输入边界名称中的特征字符，然后用鼠标点击 Match（匹配）按钮，就可以完成边界选择操作。

类似于互联网搜索引擎中的使用方法，符号*和？为通配符。如果输入 wall*，则所有名字开始为 wall 的边界将被自动选择（如 wall-1、wall-top 等）。如果这些边界已经被选择，则该操作将使这些选择被取消。如果用户输入 wall?，则所有名字中包含 wall，且 wall 后面只有一个字符的边界将被选择（如果这些边界已经被选择，则该操作将使这些选择被取消）。

3．单击 compute（计算）按钮，在 Results（结果）列表框中将显示所选择边界区域的流量计算结果，并且在 Results（结果）列表框下面的窗体中显示参与计算的所有区域流量的和。

13.2 边界上作用力的计算

在 FLUENT 中可以计算和报告指定方向上的作用力和以指定点为参考点的力矩。该功能可以用来计算升力系数、阻力系数、力矩系数等空气动力学系数。

13.2.1 计算作用力和力矩

将每个边界网格面上的压力和粘性力与指定方向的方向矢量做点积，然后再将所有边界网格上的点积求和，就可以得到边界上的作用力。除了计算压力、粘性力和合力外，还可以利用 Reference Values（参考值）面板中的参考值计算作用力系数。作用力系数被定义为作用力与$\frac{1}{2}\rho v^2 A$的商，其中ρ、v、A是在 Reference Values（参考值）输入框中给定的密度、速度和面积值。

力矩是通过将每个网格面上的作用力绕力矩中心的力矩求矢量和得到的。除了压力、粘度和力矩分量之外，还可以计算力矩系数。力矩系数定义为力矩与参考动压、参考面积和参考长度的商。

所有被选择壁面区域上的压力、粘性力、合力和力矩都将以有量纲形式和无量纲系数形式显示在主窗口中。

为了减少截断误差，在压力计算中引入一个参考压强用于处理计算网格上的压力。这个参考压强也在 Reference Values（参考值）面板中定义。例如，净压力矢量等于所有网格面作用力矢量之和，其具体计算公式则为：

$$\begin{aligned}\vec{F}_p &= -\sum_{i=1}^{n}(p - p_{ref})A\hat{n} \\ &= -\sum_{i=1}^{n} pA\hat{n} + p_{ref}\sum_{i=1}^{n} A\hat{n}\end{aligned} \qquad (13\text{-}1)$$

式中 n 为网格面总数，A 为网格面的面积，$\hat{n}$ 为网格面的单位法向量。

引入参考压强时得到的附加项，在封闭区域的计算中被相互抵消，在开放区域的计算中则多出一个力的增量，该增量的大小为开放边界的投影面积与参考压力的乘积。

13.2.2 生成一个作用力或力矩报告

使用 Force Reports（作用力报告）面板，可以算出某个边界在指定方向上受到的作用力，或绕指定点的力矩，如图 13-2 所示。启动这个面板的菜单操作如下：

Report->Forces

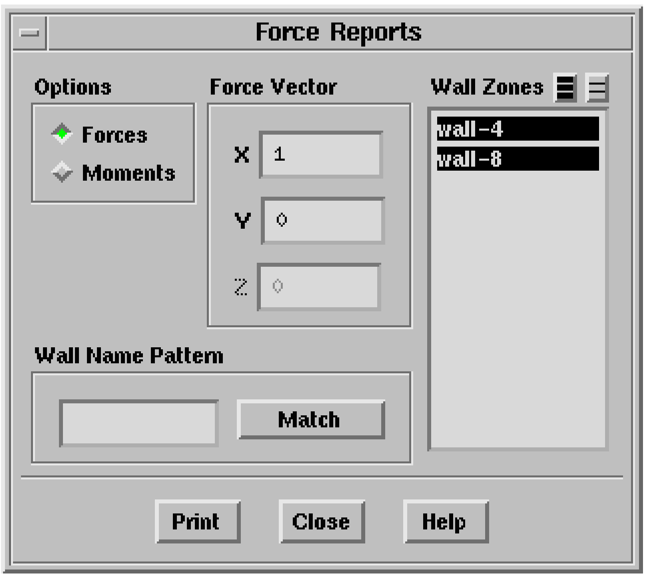


图 13-2　Force Reports 面板

生成作用力报告的具体步骤如下：

1．在 Options（选项）下选择 Forces（作用力）或 Moments（力矩）设定计算内容。

2．若选择生成作用力报告，则需要在 Force Vector（作用力矢量）中指定作用力方向的 X、Y 和 Z 分量；若选择生成力矩报告，则需要在 Moment Center（力矩中心）中指定力矩中心的 X、Y 和 Z 坐标。

3．在 Wall Zones（边界区域）列表中选择需要计算力和力矩的边界。

类似于边界流量报告中的选择方法，如果需要选择多个边界区域，可以用边界名称样式（Wall Name Pattern）选择，其使用方法与 13.1.1 中边界名称样式的使用方法相同，这里不再重复。

4．单击 Print（输出）按钮。在操作台窗口（文本窗口）中将显示出对于已选择的边界沿着指定方向的作用力矢量或绕指定力矩中心的压力、粘性力、合力或力矩，以及压力系数、粘性力、合力或力矩的系数。在报告末尾是所有指定边界上的力和力矩及与它们相关的系数。

13.3 计算投影面积

可以用 Projected Surface Areas（投影面积）面板（见图 13-3）计算指定的面沿 X、Y 或 Z 轴方向（例如在 YZ、XZ 或 XY 平面上）的投影面积。启动这个面板的菜单操作为：

Report -> Projected areas

计算投影面积的过程如下：

1. 选择投影方向（X、Y 或 Z）。
2. 在 Surfaces（表面）列表中选择准备计算其投影面积的面。
3. 设置 Min Feature Size（最小特征尺寸），用来指定面中最小的几何构形的长度（如果用户不能确定最小的几何特征的尺寸，也可以使用软件中设定的缺省值）。
4. 单击 Compute（计算）按钮，面积值将出现在 Area（面积）框和控制台窗口中。
5. 为了改善面积计算的精确度，用户可以降低 Min Feature Size 到原来值的一半再计算。重复这个过程直到计算出的面积值不再改变。注意不要调整得过小，因为那样会占用大量的内存。

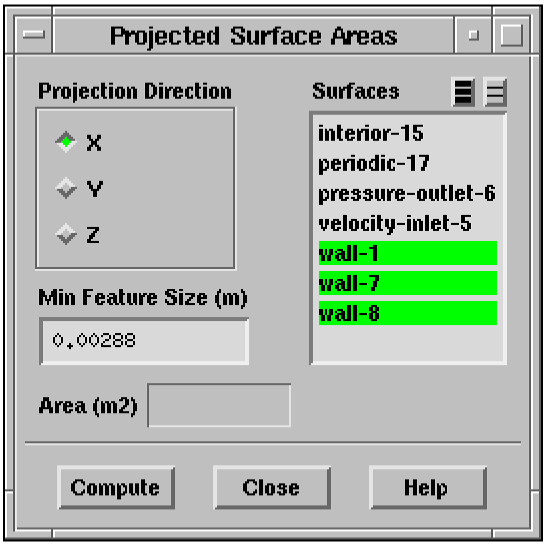


图 13-3 Projected Surface Areas（投影面积）面板

需要注意的是，这项计算仅在三维情况下可以使用。

13.4 表面积分

在计算域中特定的面上可以计算面积、质量流量和流场变量的积分、面积加权平均、质量加权平均、总和、单元面平均、单元面上的最大值和最小值、顶点平均、顶点最小值、顶点最大值等等。面（surface）是数据点的集合，这些面可能是在计算模型中创建的，也可能是用户在后处理过程中人工创建的。

由于面可以被任意地放置在流场中，因此每个数据点的变量值都是由网格节点上的值通过线性插值得到的。一些变量在网格节点上的值，可以由求解器直接计算得出，而另外一些变量在网格节点上的值，则是通过对网格中心处的值做平均计算得到的。对网格节点上的值进行这样的处理，可能使最终出现在计算报告中的结果带有一些误差。

下面是使用几种表面积分进行结果处理的例子：

1. 面积：用户可以计算速度入口区域的面积，然后用质量流量算出速度，公式如下：

$$v = \frac{\dot{m}}{\rho A} \qquad (13\text{-}2)$$

2. 面积加权平均：用户可以算出流场变量在一个固体表面上的平均值，例如可以计算一个恒温壁面上的平均热流量。

3. 质量平均：用户可以算出流场变量在一个流体表面上的平均值，例如可以计算速度入口处的平均热焓。

4. 质量流量：用户可以算出速度入口的质量流量，然后用公式（13-2）算出速度。

5. 流量：可以通过计算焓的流量得到热量流量。

6. 积分：用户可以对面上的流场变量进行积分。

13.4.1 计算面积分

1. 面积

将组成表面的每个单元面的面积相加就得到一个面的面积。单元面的形状是三角形或四边形。面积计算公式如下：

$$\int dA = \sum_{i=1}^{n} |A_i| \tag{13-3}$$

式中 A 为面积，下标 i 为单元编号，n 为单元总数。

2. 积分

将每个单元面上的面积和变量的乘积进行求和，就可以得到该变量在一个表面上的积分。每个单元面都与计算域中的某个网格相关。如果单元面是网格与等值面相交产生的，那么变量在这个单元面上的值，就是变量在这个网格上的值。如果单元面在边界上，则用插值得到的结果进行面积分，而不是直接使用网格上的变量值。这样做既可以改善计算精度，又可以保证计算结果与边界条件相一致，并且与计算报告中边界上的通量相吻合。上述积分方法可以用公式（13-4）表达如下：

$$\int \phi dA = \sum_{i=1}^{n} \phi_i |A_i| \tag{13-4}$$

式中 ϕ 为任意变量，其他符号含义与式（13-3）中相同。

3. 面积加权平均

变量的面积加权平均，等于将该变量与单元面面积的乘积求和后，在除以总面积得到的商，其公式如下：

$$\frac{1}{A}\int \phi dA = \frac{1}{A}\sum_{i=1}^{n} \phi_i |A_i| \tag{13-5}$$

式中符号含义与式（13-3）中相同。

4. 流量

变量ϕ通过一个表面的流量，等于每个单元面的面积矢量与当地速度矢量的点乘，再乘以密度和变量ϕ的当地值，最后将每个单元面上的乘积相加得到的，用公式可以表示如下：

$$\int \phi\rho\vec{v}\cdot d\vec{A}=\sum_{i=1}^{n}\phi_i\rho_i\vec{v}_i\cdot\vec{A}_i \tag{13-6}$$

式中ρ为密度，v为速度，其他符号含义与式（13-3）中相同。

5. 质量流量

将公式（13-6）中的变量ϕ取为 1，就得到质量流量的计算公式：

$$\int \rho\vec{v}\cdot d\vec{A}=\sum_{i=1}^{n}\rho_i\vec{v}_i\cdot\vec{A}_i \tag{13-7}$$

6. 质量加权平均（Mass-Weighted Average）

用公式（13-6）与（13-7）相除，就得到变量ϕ的质量加权平均公式：

$$\frac{\int \phi\rho\left|\vec{v}\cdot d\vec{A}\right|}{\int \rho\left|\vec{v}\cdot d\vec{A}\right|}=\frac{\sum_{i=1}^{n}\phi_i\rho_i\left|\vec{v}_i\cdot\vec{A}_i\right|}{\sum_{i=1}^{n}\rho_i\left|\vec{v}_i\cdot\vec{A}_i\right|} \tag{13-8}$$

7. 求和

变量在一个表面上的和等于该变量在各个单元面上数值的和，即等于：

$$\sum_{i=1}^{n}\phi_i \tag{13-9}$$

8. 单元面平均（Facet Average）

变量的单元面平均指的是用总的面数n去除该变量在所有单元面上数值的和：

$$\frac{\sum_{i=1}^{n}\phi_i}{n} \tag{13-10}$$

9. 单元面最小值（Facet Minimum）

在一个表面上，变量的单元面最小值是指该变量在该表面所有单元面上的最小值。

10. 单元面最大值（Facet Maximum）

在一个表面上，变量的单元面最大值是指该变量在该表面所有单元面上的最大值。

11. 节点平均值

变量的节点平均值等于所有节点上该变量值之和除以总的节点数 n，即等于：

$$\frac{\sum_{i=1}^{n}\phi_i}{n} \qquad (13\text{-}11)$$

12. 节点最小值

某表面上变量的节点最小值即该变量在该表面所有节点上的最小值。

13. 节点最大值

某表面上变量的节点最大值即该变量在该表面所有节点上的最大值。

13.4.2 生成一个面积分报告

使用 Surface Integrals（面积分）面板（如图 13-4），可以获得一份含有表面面积、质量流量、变量积分、变量流量、加法和、面最大值、面最小值、节点最大值、节点最小值，或质量加权平均、面积加权平均、单元面平均和顶点平均等变量的平均值等数据在内的计算报告。启动该面板的菜单操作如下：

Report -> Surface Integrals

生成报告的步骤如下：

1．在 Report Type（报告类型）中选择 Area（面积）、Integral（积分）、Area-Weighted Average（面积加权平均）、Flow Rate（变量流量）、Mass Flow Rate（质量流量）、Mass-Weighted Average（质量加权平均）、Sum（加法和）、Facet Average（单元面平均）、Facet Minimum（单元面最小值）、Facet Maximum（单元面最大值）、Vertex Average（节点平均值）、Vertex Minimum（节点最小值）或 Vertex Maximum（节点最大值）等准备计算的参数。

2．如果需要计算的是面积或质量流量报告，则忽略此步，直接进入下一步。否则在 Field Variable（流场变量）中选择用户在表面积分中准备使用的场变量：首先在上面的下拉列表中选择变量的种类，然后在下面的下拉列表中确定最终计算的变量。

3．在 Surfaces（面）列表中选择表面积分所使用的面。

如果准备选择几个类型相同的面，可以通过在 Surface Types（面类型）中选择面类型来实现——所有属于被选定类型的面将被自动选择。如果某个面已经被选择的话，则该操作将对取消对它的选择。

当然对于面的选择也可以使用 Surface Name Pattern（面名称样式）进行，其使用方法与 13.1.1 中边界名称样式的使用方法相同，这里不再重复。

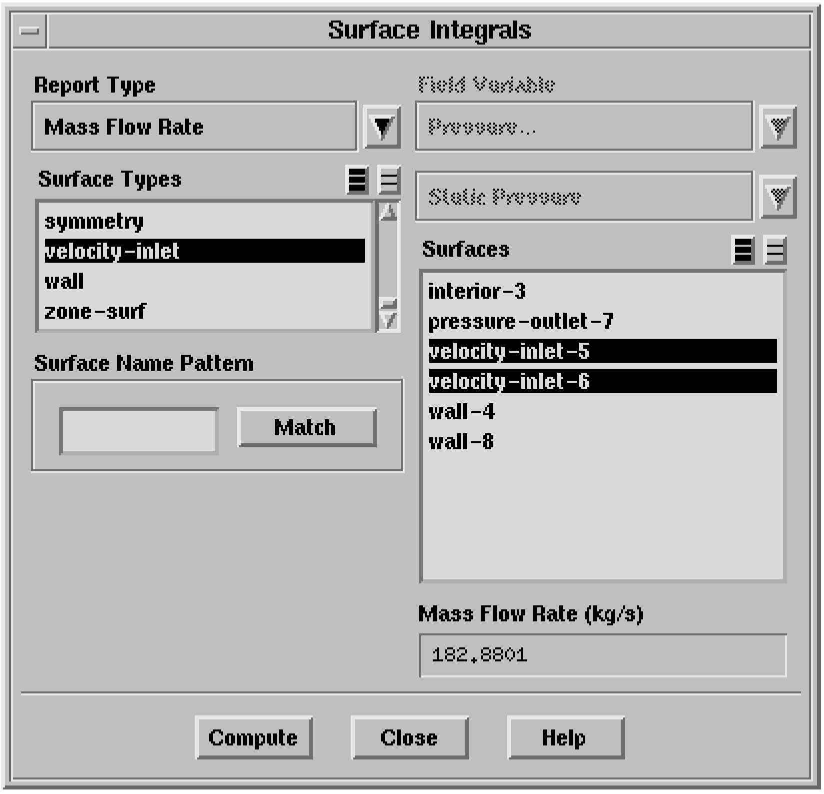


图 13-4 面积分（Surface Integrals）面板

4．单击 Compute（计算）按钮。根据用户选择的不同，结果显示栏上方的名称将调整为 Area（面积）、Integral（积分）、Area-Weighted Average（面积加权平均）、Flow Rate（变量流量）、Mass Flow Rate（质量流量）、Mass-Weighted Average（质量加权平均）、Sum（加法和）、Facet Average（单元面平均）、Facet Minimum（单元面最小值）、Facet Maximum（单元面最大值）、Vertex Average（节点平均值）、Vertex Minimum（节点最小值）或 Vertex Maximum（节点最大值）等。

下面是一些需要注意的地方：

（1）质量加权平均值更接近于高速度区域的值，即有更多质量流过的面上的变量值。

（2）使用 Surface Integrals（面积分）面板报告的流量，不如从 Flux Reports（通量）面板中得到的结果准确。

（3）单元面平均值和节点平均值最好用在面积为零的表面上。

13.5 体积分

用户可以计算某个区域的体积，并可以计算该区域上某个场变量的加法和、体积积分、体积加权平均、质量积分和质量加权积分等指标。

下面是使用不同类型的体积分报告的例子：

1. 体积：用户可以计算一个流体区域的总体积。

2. 加法和：用户可以对弥散相的质量或者能量源项进行求和，以便判断弥散相的质量或能量的净生成量，也可以计算用户自定义的质量或能量源项的净生成量。

3. 体积分：对于那些用单位体积计算的量，用户可以通过体积分得到它们的总量，比如通过对密度积分而得到质量。

4. 体积加权平均：用户可以计算质量源项、能量源项或弥散相的质量、能量交换的体积平均。

5. 质量积分：用户可以通过积分某种化学组元的质量浓度得到该组元的总质量。

6. 质量加权平均：通过质量加权平均，用户可以得到一个流动区域内变量的平均值，例如平均温度。

13.5.1 计算体积分

1. 体积

将组成某区域的所有网格的体积相加即可得到该区域的总体积：

$$\int dV = \sum_{i=1}^{n} |V_i| \qquad (13\text{-}12)$$

式中 V 为体积，n 为网格总数。

2. 加法和

某个区域上指定场变量的和是通过将该区域中每一个网格上该变量的值相加得到的：

$$\sum_{i=1}^{n} \phi_i \qquad (13\text{-}13)$$

这里符号 ϕ 仍然代表任意一种场变量。

3. 体积分

体积分是通过将网格体积与场变量的乘积相加得到。

$$\int \phi dV = \sum_{i=1}^{n} \phi_i |V_i| \qquad (13\text{-}14)$$

4. 体积加权平均

一个量的体积加权平均是将场变量与网格体积相乘，然后再将乘积求和，之后再除以总的体积得到，可用公式表达如下：

$$\frac{1}{V}\int \phi dV = \frac{1}{V}\sum_{i=1}^{n} \phi_i |V_i| \qquad (13\text{-}15)$$

5. 质量加权积分

质量加权积分是通过将密度、网格体积和场变量的乘积相加得到：

$$\int \phi\rho dV = \sum_{i=1}^{n} \phi_i \rho_i |V_i| \qquad (13\text{-}16)$$

6. 质量加权平均

质量加权平均值的计算方法可用公式表达如下：

$$\frac{\int \phi\rho dV}{\int \rho dV} = \frac{\sum_{i=1}^{n} \phi_i \rho_i V_i}{\sum_{i=1}^{n} \rho_i V} \qquad (13\text{-}17)$$

等式右边为相除关系，其中分子为选择的场变量乘以密度和网格体积，然后再求和；分母为密度和网格体积的乘积之和。

13.5.2 生成体积分报告

使用 Volume Integrals（体积分）面板，可以获得区域的体积或者指定变量的加法和、体积积分、体积加权平均、质量加权积分或质量加权平均。启动该面板的菜单操作如下：

Report -> volume integrals

生成体积分报告的步骤如下：

1．在 Options（选项）下可以选择 Volume（体积）、Sum（加法和）、Volume Integral（体积分）、Volume-Average（体积平均）、Mass Integral（质量积分）或者 Mass-Average（质量平均）来指定准备计算的内容。

2．如果只希望计算区域的体积，则忽略此步，直接进入下一步，否则在 Field Variable（场变量）下拉列表中选择参与计算的场变量。首先，在上面的下拉列表中选择场变量的种类，然后从下面的列表中确定最后计算的变量。

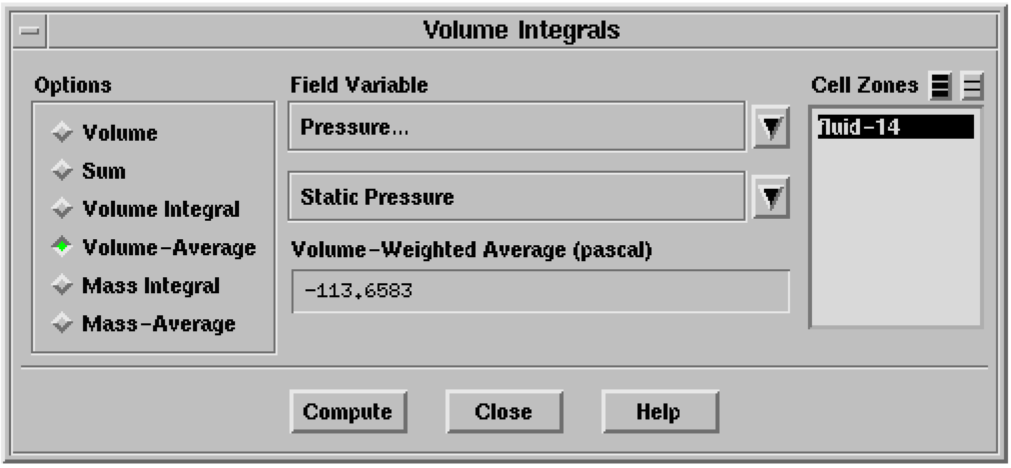


图 13-5 Volume Integralsl（体积分）面板

3．在 Cell Zones（网格区域）列表中选择将要进行计算的区域。

4．单击 Compute（计算）按钮。根据用户选择的不同，结果显示栏上面的名称将自动调整为 Total Volume（总体积）、Sum（加法和）、Total Volume Integral（总体积积分）、Volume-Weighted Average（体积加权平均）、Total Mass-Weighted Integral（总质量加权平均）或 Mass-Weighted Average（质量加权平均）等。

13.6 柱状图报告

在 FLUENT 中，用户可以在控制台窗口（文本窗口）中以柱状图格式打印出几何和结果数据，或者在图形窗口中画出一个柱状图。柱状图的图形显示和定义过程参见后处理技术相关章节的内容。

报告中将显示网格的数目、选定变量或函数的取值范围和占内部网格总数的百分比等信息，下面是一个例子：

0 cells below 1.195482 (0 %)
2 cells between 1.195482 and 1.196048 (4.1666667 %)
1 cells between 1.196048 and 1.196614 (2.0833333 %)

0 cells between 1.196614 and 1.19718 (0 %)
0 cells between 1.19718 and 1.197746 (0 %)
2 cells between 1.197746 and 1.198312 (4.1666667 %)
1 cells between 1.198312 and 1.198878 (2.0833333 %)
6 cells between 1.198878 and 1.199444 (12.5 %)
9 cells between 1.199444 and 1.20001 (18.75 %)
25 cells between 1.20001 and 1.200576 (52.083333 %)
2 cells between 1.200576 and 1.201142 (4.1666667 %)
0 cells above 1.201142 (0 %)

用户可以用 Solution Histogram（解的柱状图）面板生成一个这样的柱状图，启动该面板的菜单操作为：

Report->volume histogram

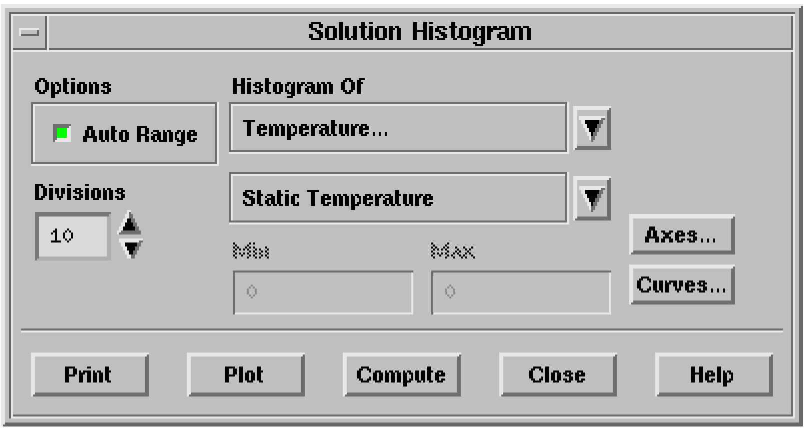


图 13-6 Solution Histogram（解的柱状图）面板

要生成这样一份报告，应该点击 Print（文字输出）按钮，而不要按 Plot（绘图）按钮。

13.7 参考值设定

用户可以控制参考值的设定，这些参考值用于导出物理量和无因次系数的计算，并且仅用于后处理过程。

下面是一些使用参考值的例子：

1. 作用力系数计算需要使用参考面积、密度和速度，而计算压力还要使用参考压强。
2. 力矩系数计算需要使用参考长度、面积、密度和速度。
3. 雷诺数计算需要使用参考长度、密度和粘度。

4. 计算压力和总压系数需要使用参考压强、密度和速度。
5. 计算熵需要使用参考密度、压力和温度。
6. 计算表面摩擦系数需要使用参考密度和速度。
7. 计算热传导系数需要使用参考温度。
8. 计算涡轮机效率需要使用比热比。

13.7.1 设定参考值

参考值的设定在 Reference Values（参考值）面板（如图 13-7）中进行，启动该面板的菜单操作为：

Report -> reference values

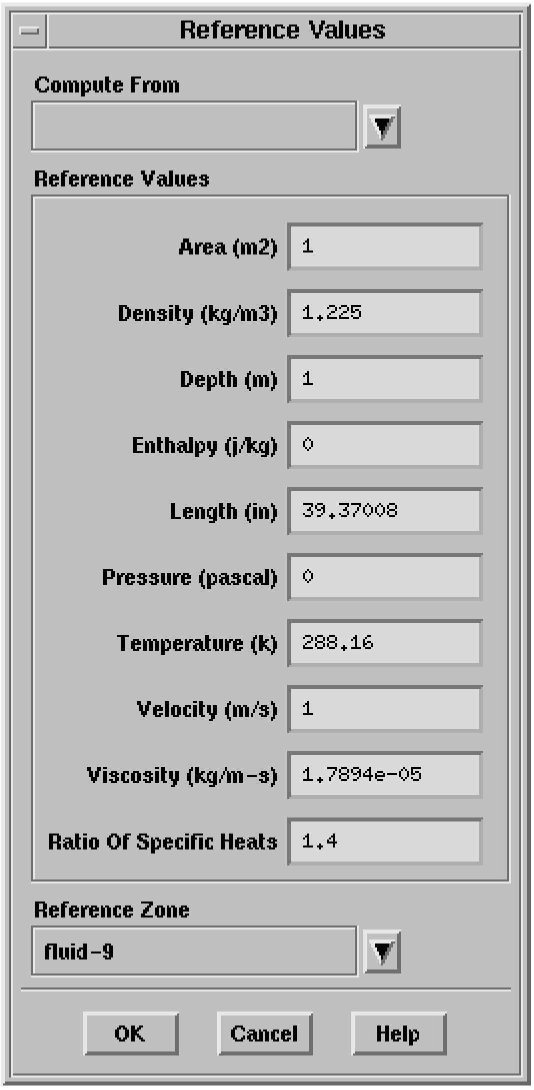


图 13-7 Reference Values（参考值）面板

可以人工输入参考值或者以边界上的物理量为基准计算参考值，可以被设定的参考值有 Area（面积）、Density（密度）、Enthalpy（焓）、Length（长度）、Pressure（压强）、Temperature（温度）、Velocity（速度）、dynamic Viscosity（动力粘度）和 Ratio Of Specific Heats（比热比）。对于二维问题，还可以定义 Depth（深度）。这个值用于计算通量和作用力，其单位是单独设定的，与 Set Units（单位设置）面板中设置的长度单位无关。

如果用户想根据某个边界条件计算参考值，需要在 Compute From（从……计算）下拉列表中选择该边界区域的名称。需要注意的是，这种获得参考值的方法仅能用于部分参考值的设置。比如参考长度和面积就不能从边界条件中导出，用户仍然需要人工设定这些值。

在使用人工方式设定参考值时，用户只要简单地在参考值（Reference Values）旁边的输入栏中键入每个值即可。

13.7.2 设定参考区域

如果计算模型中使用了多个参考坐标系，或者在计算中使用了滑移网格，则可以画出相对于指定参考区域的速度和其它相关量。在 Reference Zone（参考区域）下拉列表中选择期望的区域，改变参考区域可以使用户获得相对于不同区域的速度值（还有总压、温度等）。

13.8 关于算例设置的摘要报告

在计算过程中有时需要查阅算例设置，包括物理模型、边界条件、材料属性和求解控制参数等等。在 FLUENT 中可以将所有这些设置放在一份报告中，这样就大大减少了在大量面板中逐个检查算例设置情况的麻烦。

13.8.1 生成一个摘要报告

用 Summary（摘要）面板可以得到一个关于算例设置的摘要报告（见图 13-8），启动该面板的菜单操作如下：

Report -> Summary...

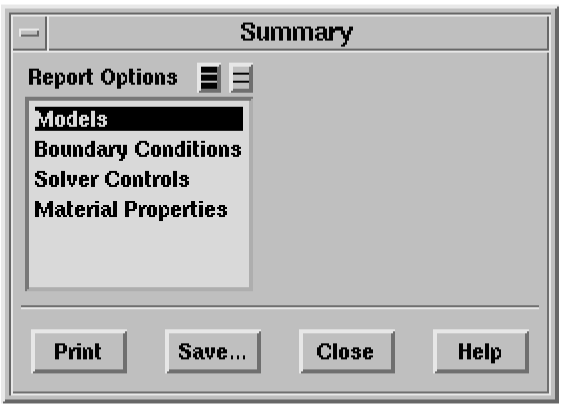


图 13-8 Summary（摘要）面板

生成一个摘要报告的步骤如下：

1．在 Report Options（报告选项）列表中选择用户想在报告中看到的信息，可供选择的项目包括 Models（模型）、Boundary Conditions（边界条件）、Solver Controls（求解器控制参数）和 Material Properties（材料性质）。

2．如果单击 Print（文本输出）按钮，则信息在 FLUENT 的控制台窗口里面输出；如果单击 Save（保存）按钮，并在 Select File（文件选择）窗口中输入文件名，则信息将被保存在一个文本文件中。

第十四章 TECPLOT 简介

TECPLOT 是 Amtec 公司推出的一个功能强大的科学绘图软件，它不仅可以绘制函数曲线、二维图形，而且可以进行三维面绘图和三维体绘图，并提供了多种图形格式，同时界面友好、易学易用。TECPLOT 有针对 FLUENT 软件的专用数据接口，可以直接读入算例文件和数据文件，也可以在 FLUENT 软件中选择面和变量，然后直接输出 TECPLOT 格式文档。现在 TECPLOT 软件的最新版本为 9.2 测试版，本章将根据 9.0 版进行介绍。

14.1 TECPLOT 基本功能

本章将介绍 TECPLOT 的用户界面，同时利用 TECPLOT 软件包中自带的数据样本，介绍各种图形的创建过程。

14.1.1 TECPLOT 软件的启动

在 Windows 操作系统中，启动 TECPLOT 软件可以从“开始”按钮开始，或者双击桌面上的快捷方式直接启动。如果选择从“开始”按钮启动，其操作过程如下：

开始 -> 程序 -> TECPLOT9.0 -> TECPLOT

启动后首先出现的是启动界面，随后就是 TECPLOT 窗口，如图 14-1 所示：



图 14-1 Windows 操作系统下 TECPLOT 界面

14.1.2 界面

图 14-2 是在没有加载任何数据的情况下，TECPLOT 的开始界面。界面可以分成四部分，即菜单栏、工具栏、工作区和状态栏。

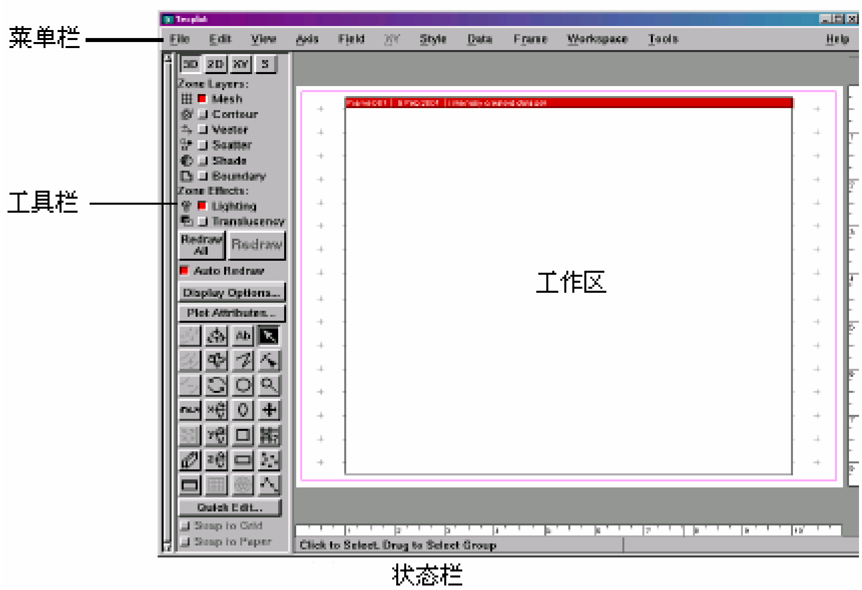


图 14-2 TECPLOT 界面上的分区

1. 菜单栏

如图 14-3 所示，通过菜单栏可以使用绝大多数 TECPLOT 的功能，它的使用方式类似于一般的 Windows 程序，是通过对话框或者二级窗口来完成的。



图 14-3 菜单栏

TECPLOT 的功能包含在如下菜单中：

（1）File（文件）：在这个菜单中进行文件和图形的读写、打印与输出，记录或运行宏，设置并保存配置，以及退出系统等操作。

（2）Edit（编辑）：进行图形对象的剪切、复制、粘贴、清除、改变显示顺序、修改数据点等操作。TECPLOT 的剪切、复制和粘贴只在 TECPLOT 内部有用。如果想与其他 Windows 程序交换图形，可以用 Copy Plot to Clipboard（将图形拷贝到剪贴板）功能。

（3）View（视图）：用来控制视图显示，包括比例缩放、数据范围、三维旋转等，还可以在帧之间进行图形的复制和粘贴操作。

（4）Axis（坐标轴）：控制 XY、二维、三维等帧模式下的坐标轴显示形式。

（5）Field（场）：用来控制二维、三维帧模式中场变量的绘制，包括网格、等值线、矢量、阴影、流线、三维等值面、三维切片和边界曲线等。

（6）XY：在这里可以控制 XY 曲线的绘制。

（7）Style（风格）： 用来控制文本、几何体（多线、园、矩形、椭圆、正方形）、数据标签、空格等功能，还可以复制、粘贴风格表单文件。

（8）Data（数据）：用来创建、操作、检查数据，比如创建简单区域、插值、三角形剖分，以及用类似 FORTRAN 的语句创建和修改变量。

（9）Frame（帧）：创建、编辑、控制帧。

（10）Workspace（工作区）：用来控制工作区的属性，包括色彩图板、页面网格、显示选项和标尺等等。

（11）Tools（工具）：用来快速运行已经制作好的宏、创建简单的动画，软件包中附带的一些功能（add-on）在安装完毕后也在这里调用，同时可以快速编辑图形标记。

（12）Help（帮助）：打开帮助文档。

2. 工具栏

TECPLOT 工具栏将常用的画图控制用按钮形式集中在工具栏中，同时还可以选择帧模式、激活图形层和采用快照模式。工具栏如图 14-4 所示。

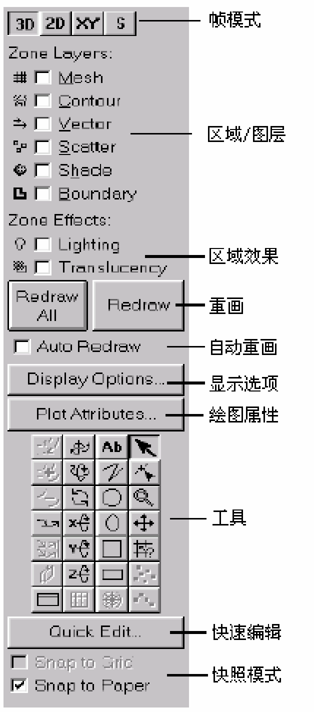


图 14-4 工具栏

（1）帧模式

帧模式即当前帧内所绘制图形的类型，其中包括：

1）三维：绘制三维面或者体。

2）二维：绘制二维图形。

3）XY：绘制 XY 曲线，即常见的函数关系曲线。

4）S （草图）：在没有数据的情况下随意绘制的图形，例如草图、流程图等。

图形是由帧模式、数据集、活动图形层及其属性定义构成的。每个帧模式都只能显示数据的一种特征。

（2）区域层与图形层

区域层（Zone Layer）是表达数据集合的一种方式。完整的图形应该包括所有被激活的层、坐标轴、文本、几何对象和在绘图期间加入的其他图形元素。在二维和三维帧模式下共有 6 种区域层；在 XY 帧模式下有 4 种图形层；草图模式中则没有使用层概念。

1）二维和三维帧模式下的 6 种区域层。

Mesh（网格）：在网格层中，用线连接数据点。

Contour（等值线）：在等值线层中，可以用曲线形式或填充形式绘制等值线，也可以同时采用等值线和填充方式进行绘制。

Vector（矢量）：在矢量层中，绘制带方向和大小的矢量图形。

Scatter（散点）：在散点层中，在每一个数据点的位置上绘制符号。

Shade（阴影）：在阴影层中，可以为指定区域加阴影，或用添加光源的方式为三维曲面添加阴影。与光照（Lighting）效果配合使用，可以制作平面阴影或交错阴影；与透明（Translucency）效果配合使用，可以创建透明曲面。

Boundary（边界）：在边界区域层中，为有序数据集合绘制区域边界。

2）XY 模式下的 4 种图形层，如图 14-5。

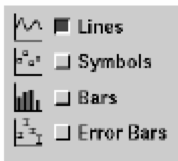


图 14-5 XY 图形态图

Lines（线状图）：用线段或拟合曲线形式，绘制变量（X，Y）的函数关系曲线。

Symbols（符号图）：用散点形式表达变量（X，Y）的函数关系。

Bars（柱状图）：用水平或垂直柱状图形式，表达变量（X，Y）的函数关系。

Error Bars（误差柱状图）： 可以用几种格式绘制误差柱状图。

（3） 区域效果（Zone Effects）

在三维帧模式下会出现如图 14-6 的选择框。选择框内的 Lighting（光照效果）和 Translucency（透明效果）只对着色和带阴影效果的等值面起作用。

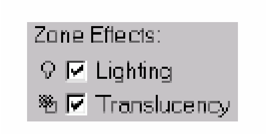


图 14-6 区域效果选项

（4） 重画按钮（redraw button）

除非选择自动重画（automatically redraw），否则每次对图形进行改动后都需要用重画按钮手动更新。重画按钮如图 14-7 所示：点击 Redraw（重画）按钮重画当前帧，点击 Redraw All（全部重画）按钮，重画全部帧；按下 shift 键，同时点击 Redraw All（全部重画）按钮，将重新生成工作区。

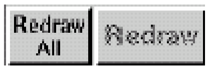


图 14-7 重画按钮

（5）自动重画（Auto Redraw）：选择这个选项将连续不断地自动更新图形内容。

（6）显示选项按钮（Display Option Button）：点击这个按钮可以启动 Display Option（显示选项）对话框。在这个对话框中可以设定 TECPLOT 的状态栏和性能参数。

（7）绘图属性按钮（Plot Attributes Button）：点击这个按钮可以打开 Plot Attributes（绘图属性）对话框，进行区域外观设置。

（8）工具按钮（tool button）：每个工具按钮对应于一种鼠标操作，共有 28 种操作（如图 14-8），可以归纳为 12 个类，即：

1）等值线模式。

2）流线模式。

3）切片模式。

4）帧模式。

5）创建域模式。

6）三维旋转模式。

7）文字模式。

8）几何形状模式。

9）鼠标指示模式：选择与调整。

10）视图模式：缩放、移动/放大。

11）鼠标探测模式。

12）数据提炼模式。

3. 状态栏

TECPLOT 窗口底部的状态栏，在鼠标移动过工具栏、菜单和按钮时会给出帮助提示。状态栏可以在 File（文件）菜单下的 Preferences（参考）子菜单中设定。

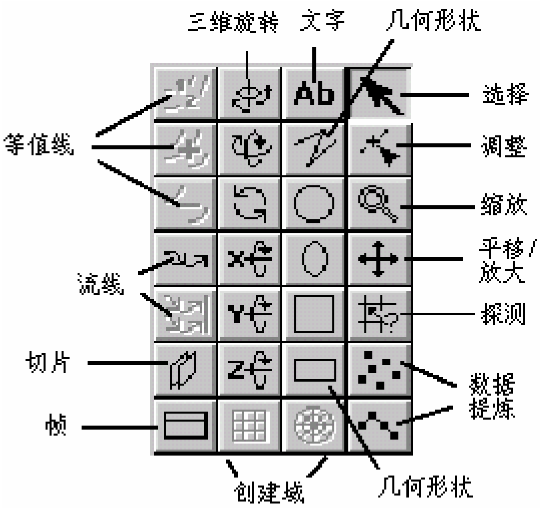


图 14-8 工具栏与鼠标模式

4. 工作区

工作区是进行绘图工作的区域。如图 14-9 所示，绘图工作都是在帧中完成的。在默认情况下，TECPLOT 显示网格和标尺。

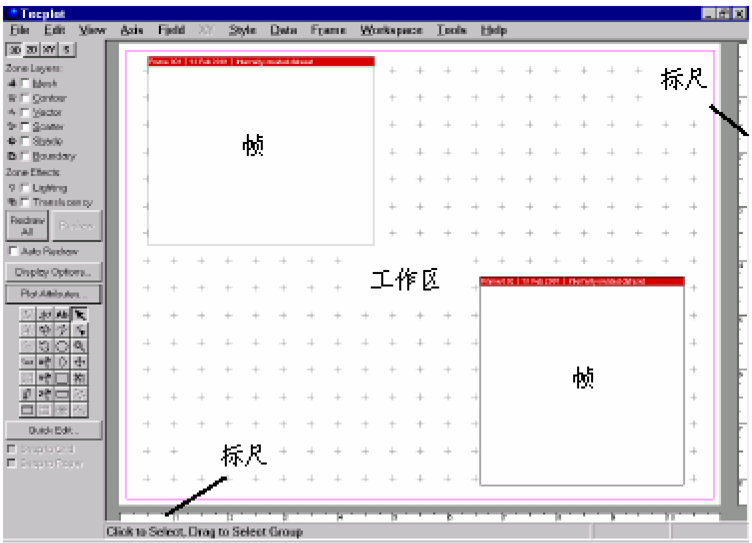


图 14-9 工作区

14.2 TECPLOT 数据格式

TECPLOT 中的数据结构可以分为两个层次（如图 14-10），即“数据集合”层和“区

域”层。“数据集合”包含一个或者多个数据“区域”和数据块。“区域”中则是具体的数据。“区域”既可以从数据文件中读入，也可以在 TECPLOT 中直接创建。

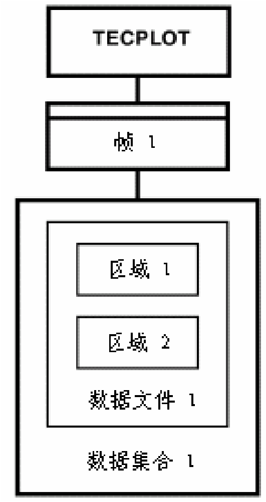


图 14-10 TECPLOT 数据结构示意图

TECPLOT 在启动时首先创建一个空的帧，然后在读入数据文件或创建区域时，再把数据集合加入到活动帧中。一个数据集合可以和多个帧相关联，如图 14-11。

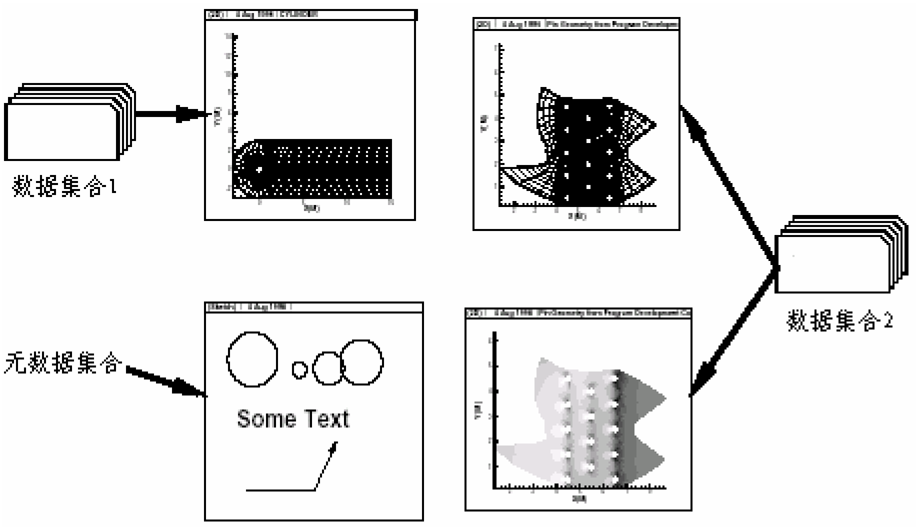


图 14-11 多帧数据连接方式

如果读入的数据文件超过一个，TECPLOT 软件会自动将数据组合为一个数据集合。如果两个帧使用相同的数据集合，则它们的标题栏颜色相同。图 14-12 给出了一个稍微复杂的 TECPLOT 数据结构：帧 1 和帧 2 使用的数据集合 1 由一个包含有 3 个区域的数据文件

构成；帧 3 使用数据集合 2， 由一个包含 2 个区域的数据文件和一个包含 3 个区域的数据文件组成；帧 4 使用数据集合 3，由一个包含 2 个区域的数据文件构成。

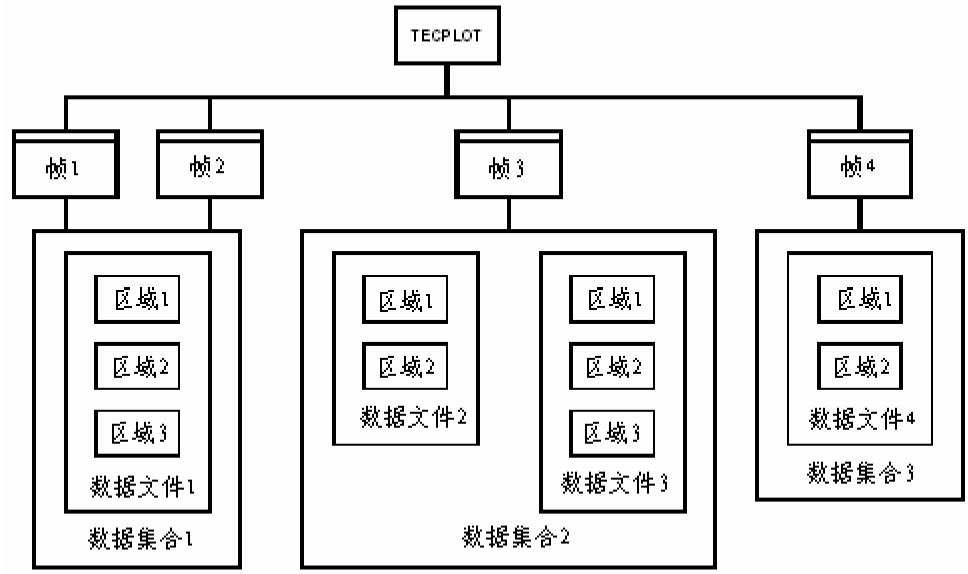


图 14-12 复杂数据结构

1. 多数据区域

多数据区域可以用来方便绘制复杂结构，或者细分结构图标。也可以用来表示不同时间步的数据，或者不同测量方法的数据。图 14-13 为一个利用多区域数据的范例。

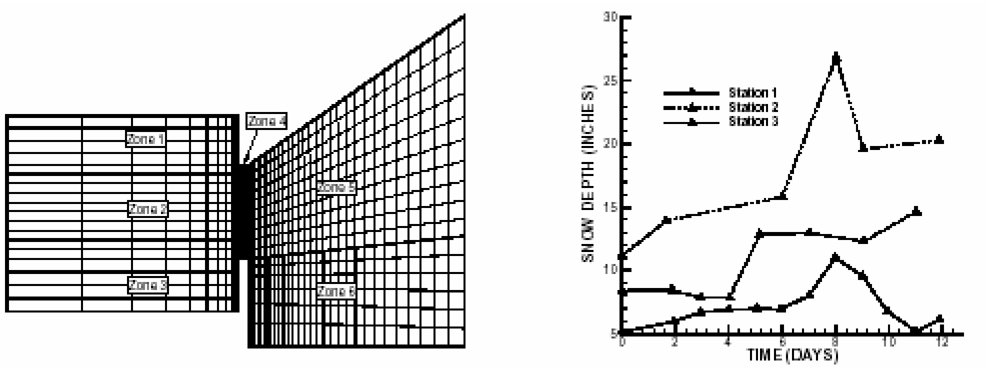


图 14-13 多数据区域应用示例

2. 数据区域中的数据结构

TECPLOT 可以使用两种数据类型：有序数据和有限元数据。

（1）有序数据

有序数据是按照逻辑顺序保存于一维、二维或者三维数组中的数据集合。TECPLOT 用 I、J、K 表示数组下标。最常见的有序数据形式为：

1）I 序列：以 I 为下标的一维数组，其中 I=1，2，3，…，Imax，如果每个点有 N 个变量（V1，V2，V3，⋯，VN），则数组如表 14-1 所示：

表 14-1 I 序列数据表（适用于 XY 曲线绘图）

V1	V2	V3	V4	V5	V6	. . .	V*N*	(Values at data point *I* = *1*.)
V1	V2	V3	V4	V5	V6	. . .	V*N*	(Values at data point *I* = *2*.)
V1	V2	V3	V4	V5	V6	. . .	V*N*	(Values at data point *I* = 3.)
V1	V2	V3	V4	V5	V6	. . .	V*N*	
V1	V2	V3	V4	V5	V6	. . .	V*N*	
V1	V2	V3	V4	V5	V6	. . .	V*N*	
V1	V2	V3	V4	V5	V6	. . .	V*N*	(Values at data point *I* = *IMax*.)

2）IJ 序列：以 I、J 为下标的二维数组。可以用于绘制二维或者三维曲面。每个数据点由一对参数（I，J）定义，并且具有四个相邻数据点。如图 14-14 所示：

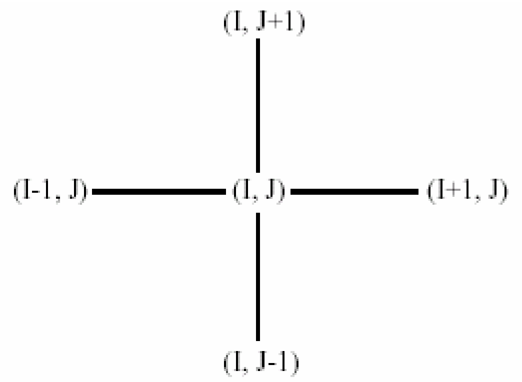


图 14-14 IJ 系列数据点相邻数据

3）IJK 序列：以 I、J、K 为下标的三维数组。每个点由三个参数（I，J，K）定义，周围有 6 个相邻数据点（边界除外），如图 14-15 所示。

上述三种排序形式是 TECPLOT 的缺省形式，但是在实际使用过程中也可以选用其他排序形式，例如以 J 为下标或者以 K 为下标的一维数组，或以 JK 或 IK 为下标的二维数组。

（2）有限元数据

有限元数据或称 FE（Finite Element）数据，是一种把数据点作为二维或者三维空间中的点，并按规律连接成单元或者网格的数据结构方法。有限元数据既定义了节点，也定义了由节点连接而成的单元。有限元数据可以分为两类，即：

1）FE-面数据：用三角形或者四边形单元定义二维场或者三维面。FE-面数据中既包含场变量数据，也包括定义节点类型和连接方式的几何信息。

2）FE-体数据：用四面体或者六面体单元定义三维场。FE-体数据中既包含场变量数据，也包含分区信息、节点类型和连接方式等几何信息。

理论上说，上述两种数据类型对数据点个数没有限制，但是由于计算机内存容量是有限的，因此实际上数据点个数也是有极限的。

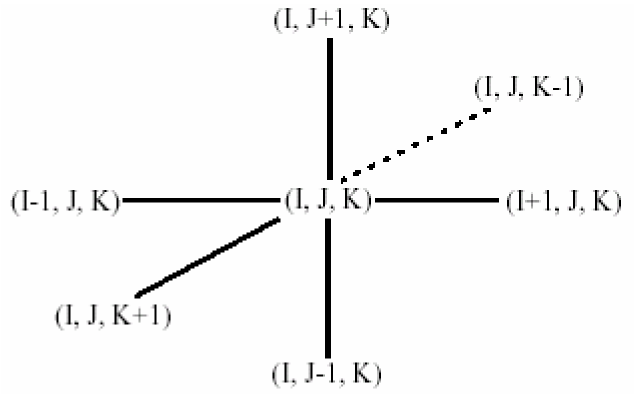


图 14-15 IJK 序列数据相邻数据

3. 查看数据集合信息

可以利用 Data（数据）菜单的子菜单 Data Set Info（数据集合信息）选项，来打开 Data Set Information（数据集合信息）窗口查看当前数据集合的信息，包括数据集合标题、分区、变量名称以及变量的最大与最小值，也可以用来改变数据集合的标题、分区、变量名称等，如图 14-16 所示：

在 zone/variable（分区和变量）标签中包含的信息如下：

（1）Zone（s）（区域）：按编号列出所有区域的名称。

（2）Zone Name（区域名称）：编辑选定区域的名称。

（3）Zone Type （区域类型）：显示选定区域的类型。对于有序数据类型，将同时显示 IMax、JMax 和 KMax。对于有限元数据类型，则显示单元类型、单元节点数和单元个数。

（4）Variables（变量）：按名称列出所有变量。

（5）Variable Name（变量名）：编辑选定变量的名称

（6）Var Type（变量类型）：显示选定变量类型。

（7）Var Range - Selected Zone（变量范围－选定区域）： 显示选定变量在选定区域内的取值范围。

（8）Var Range - Activated Zone（变量范围－激活区域）： 显示选定变量在所有激活区域内的取值范围。

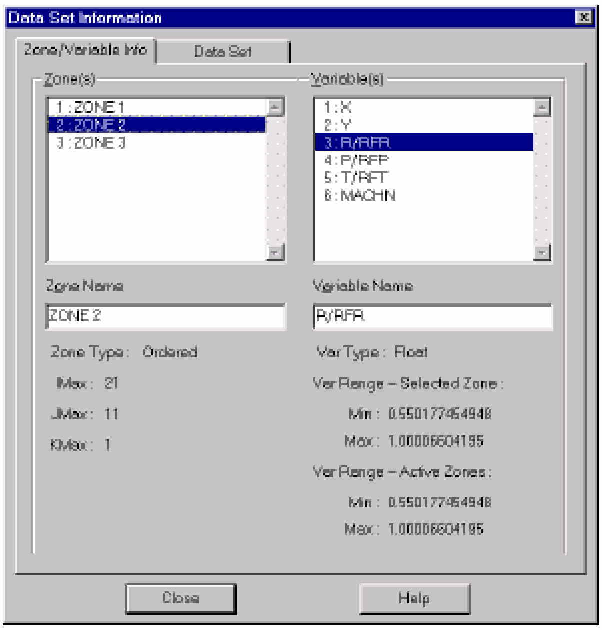


图 14-16 数据信息窗口

在 Data Set（数据集合）标签中包含下列信息：

（1）Data Set Title（数据集合标题）：输入当前数据标题，或者编辑当前数据标题。

（2）Data File（数据文件）：列出所有外部数据文件的文件名和路径。

（3）Var Load Mode（变量加载模式）：根据所用方法的不同，该项显示可以采用 By Position（根据位置）或者 By Name（根据名称）。

（4）Locked By（被……锁定）：如果数据集合被 TECPLOT 的附带模块（add-on）锁定，则可以在这里看到被哪个模块锁定。“锁定”的含义是帧或区域因与某个数据集合相结合而无法被删除。

4. ASCII 数据文件

TECPLOT 的数据文件可以是二进制格式的，也可以是 ASCII 格式的。二进制格式文件占用内存少，因此在可能的情况下，应该尽量采用二进制格式。ASCII 格式的数据文件采用 ASCII 码记录数据内容，因而可以直接阅读、检查和编辑。为了对 TECPLOT 的数据结构、数据格式有个直观印象，这里将以 ASCII 文件为范例，介绍数据文件的内容。

数据文件包括五部分，即头信息、数据记录、文本内容、几何信息和标签，其中数据记录又可以采用有序数据或有限元数据，具体内容可以用下面的表格予以概括。

表 14-1 TECPLOT 数据文件内容

数据段	数据项目	说明
文件头	TITLE = ”filetitle” VARIABLES = “vianame1” “vianame2” …	数据文件标题 变量名 1 变量,名 2 …
有序数据段	ZONE T = “zonetitle” I = Imax J = Jmax K = Kmax C = Color F = 有序数据格式 D = （duplist） DT = （datatypelist）	区域标题 I 方向最大点数 J 方向最大点数 K 方向最大点数 颜色设置，可以设置为 BLACK（黑）、RED（红）、GREEN（绿）、BLUE（蓝）、CYAN（青）、YELLOW（黄）、PURPLE（紫）、WHITE（白）、CUST1…CUST8（用户自定义）等颜色 可设为 POINT（点）或 BLOCK（块） 与上一个区域中已经出现过的变量清单 数据类型清单
有限元数据段	ZONE T = “zonetitle” N = numnodes E = numelements ET = elementtype C = color F = feformat D = （duplist）	区域标题 节点数 单元数 单元类型，可取值为 TRIANGLE（三角形）、QUADRILATERAL（四边形）、TETRAHEDRON（四面体）、BRICK（六面体）。 颜色，取值同上。 单元数据格式，可取值为 FEPOINT 或 FEBLOCK 与上一个区域中已经出现过的变量清单

	DT = （datatypelist）	数据类型清单
文本记录段	TEXT	
	X=xorigin	对象的 X 原点坐标。
	Y=yorigin	对象的 Y 原点坐标。
	F=font	可用字体，包括 HELV、HELV-BOLD、TIMES、TIMES-ITALIC、TIMES-BOLD、TIMES-ITALIC-BOLD、COURIER、COURIER-BOLD、GREEK、MATH 和用户自定义字体。
	CS=coordinatesys	文本坐标系，可以是 FRAME（帧）或 GRID（网格）。
	HU=heightunits	高度单位。在 FRAME（帧）坐标系中，单位为 FRAME 或 POINT；在 GRID（网格）坐标系中，单位为 GRID 或 FRAME。
	AN=textanchor	文本定位点。可以选择 LEFT（左）、CENTER（中）、RIGHT（右）、MIDLEFT（中左）、MIDCENTER（中心）、MIDRIGHT（中右）、HEADLEFT（上左）、HEADCENTER（上中）、HEADRIGHT（上右）。
	C=color	颜色设置，取值范围同上。
	A=angle	角度，单位为“度”，起始线为水平线，逆时针计算。
	H=height	以高度单位设置的文本高度。
	LS=linespacing	多行文本的行距。
	S=scope	范围。取值为 LOCAL（局部）或 GLOBAL（全局）。
	T="*text*"	文本内容。
	BX=boxtype	边框类型，包括 NOBOX
	BXM=boxmargin	

	BXF=boxfillcolor **BXO=**boxcolor **LT=**boxlinethickness **ZN=**zone **MFC="**macrofunctioncommand**"**	（无边框）、HOLLOW（空心边框）或 FILLED（填充边框）。 用文本高度为基数，以百分比表示的文本周围的空白间距。 边框填充颜色。 边框颜色。 边框线型宽度。 区域编号。 宏命令函数
几何记录段	GEOMETRY **X=**xorigin **Y=**yorigin **Z=**zorigin **CS=**coordinatesys **C=**color **L=**linetype **PL=**patternlength **LT=**linethickness **T=**geomtype **EP=**numellipsepts **AST=**arrowheadstyle **AAT=**arrowheadattach	对象在 X 方向的起始位置。 对象在 Y 方向的起始位置。 对象在 Z 方向的起始位置。 坐标系选择，可以取 FRAME（帧）或 GRID（网格）坐标。 颜色设置，同上。 线型设置，可以取值为：SOLID（实线），DASHED（虚线），DASHDOT（点虚线），DOTTED（点划线），LONGDASH（长虚线），DASHDOTDOT（虚点线） 指定线型的点、线长度。 几何轮廓线的线宽。 几何形状，可以取值为：LINE（线），SQUARE（正方形），RECTANGLE（矩形），CIRCLE（圆），ELLIPSE（椭圆），LINE3D（三维线）。 用于逼近圆和椭圆的点数。 箭头形式，可以取值为：

	ASZ=arrowheadsize **AAN=**arrowheadangle **DT=**datatype **S=**scope **F=**geomformat **FC=**geomfillcolor **ZN=**zone **MFC="**macrofunctioncommand**"**	PLAIN（普通），HOLLOW（空心），FILLED（实心）。 箭头附带标志，可以取值为：NONE（无），BEGINNING（起始标志），END（结束标志），BOTH（二者都有）。 箭头尺寸，使用帧单位。 箭头角度，单位为“度”。 数据类型，可以取值为：SINGLE（单精度）或 DOUBLE（双精度），仅用于多线绘制。 范围，可以取值为：LOCAL（局部）或 GLOBAL（全局）。 几何格式，可以取值为：POINT（点）或 BLOCK（块）。 几何填充色。 几何对象在区域内的编号。 宏命令。
定制标签记录段	CUSTOMLABELS "label1" "*label2*" ...	 用户自定义标签字串 1。 用户自定义标签字串 2。 ……

下面是一个 TECPLOT 数据文件实例：第 1 行为文件标题；第 2 行是 3 个变量的名称，第 4 行和第 9 行是数据分区的声明，第 5 行到第 8 行是第一个数据分区中的数据记录，每一列对应于一个变量，第 10 行到第 13 行是第二个数据分区中的数据记录，第 14 行是文本记录行，包括文本内容、位置、高度等参数。

```
TITLE = "Example: Multi-zone XY-Plot"
VARIABLES = "Position", "Temperature", "Pressure"
ZONE T="0.0 seconds", I=4
71.30 563.7 101362.5
86.70 556.7 101349.6
103.1 540.8 101345.4
124.4 449.2 101345.2
ZONE T="0.1 seconds", I=4
71.31 564.9 101362.1
84.42 553.1 101348.9
```

103.1 540.5 101344.0

124.8 458.5 101342.2

TEXT CS=GRID, HU=GRID, X=0.36, Y=0.87, H=0.04, T="SAMPLE CASE"

14.3 TECPLOT 读入 FLUENT 文件

TECPLOT 可以直接读取 10 种格式的数据文件：

（1）CGNS 格式，即计算流体力学通用记号系统（Computational Fluid Dynamics General Notation System）。

（2）DEM 格式，即数字高度图格式（Digital Elevation Map）。

（3）DXF 格式，即数字交换格式（Digital eXchange Format）。

（4）Excel 表格式（仅适用于 Windows 系统）。

（5）FLUENT 5.0 以上的数据格式，包括算例文件和数据文件。

（6）Gridgen 数据格式。

（7）HDF 格式，即层级数据格式（Hierarchical Data Format）。

（8）Image（映像）文件格式。

（9）PLOT3D 数据格式。

（10）文字表单（Text Spreadsheet）格式。

启动数据加载窗口的菜单操作步骤为：

File -> Import

数据加载窗口如图 14-17 所示。在窗口中选择 Fluent Data Loader（FLUENT 数据加载器），然后点击 OK 按钮，既可开始 FLUENT 数据加载操作。

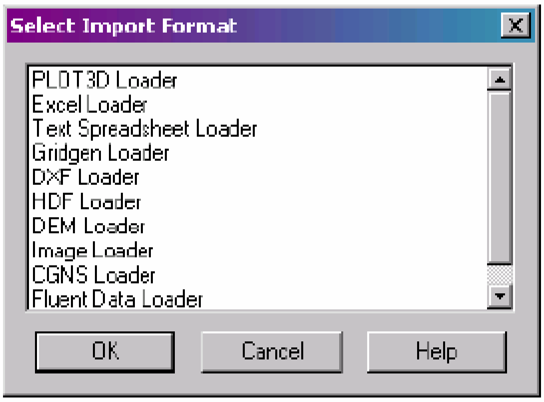


图 14-17 数据加载窗口

因篇幅所限，本节下面将讲解 FLUENT 算例文件和数据文件（即.cas 文件和.dat 文件）的加载过程，其他格式文件的加载请参考 TECPLOT 的用户手册。

由于 TECPLOT 只接受 FLUENT5.0 及以上版本的数据，所以如果需要读入早期版本的文件，就必须首先读入 FLUENT5.0 或以上版本，重新保存后再读入 TECPLOT。

因为 FLUENT 采用有限体积法进行计算，所以流场内部的计算结果都设置在控制点上，即网格体的几何中心点上，而流场边界的数值则在网格边界面的中心点上。因为 TECPLOT 要求所有的数据位于网格节点上，所以在加载数据时会采用算术平均将控制点数据换算成节点数据。

FLUENT 文件数据加载器如图 14-18 所示，其中有如下三个选项：

（1）Load Grid and Solution Data（加载网格数据和计算结果）：读入算例文件和数据文件。如前所述，网格信息存放在算例文件中，流场变量的计算结果保存在数据文件中。TECPLOT 将读入所有的变量并将其加入 TECPLOT 数据集合。如果某个变量仅出现在某个区域上，则在其他区域上，TECPLOT 会自动将该变量设置为 0。

（2）Load Grid Only（只加载网格）：只读入算例文件，并设置对应的网格选项。

（3）Load Residuals Only（只加载残差）：只从数据文件中加载残差数据（收敛史）。

在上述三个选项的下方是与这三个选项相对应的网格选项（Grid Option）组合框，其中包括如下选项：

（1）Case File（算例文件）：输入要加载的文件名，或者点击 Select（选择）按钮，直接从硬盘上选择算例文件。

（2）Load Cells and Boundaries（加载单元和边界）：从算例文件中读入单元数据和边界区域。每个流体、固体或边界区域在 TECPLOT 软件中将被显示为独立的区域。

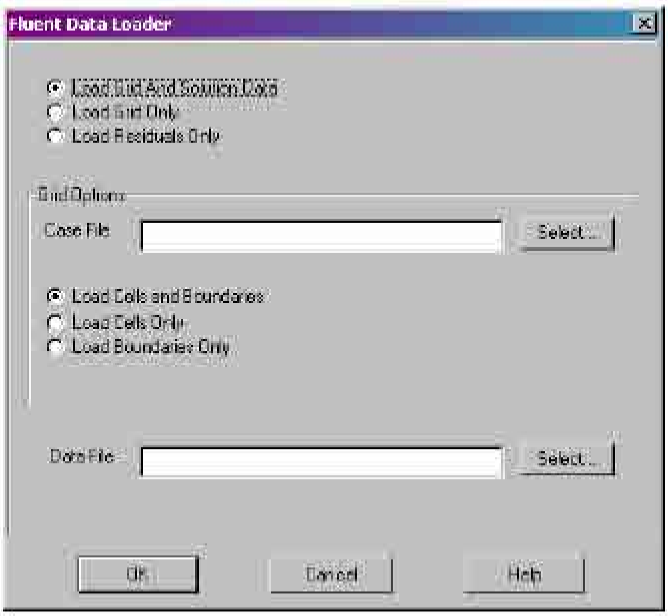


图 14-18 FLUENT 数据加载器

（3）Load Cells Only（只加载单元数据）：只加载单元数据区域。每个区域在 TECPLOT

中都显示为独立的数据区域。

（4）Load Boundaries Only （只加载边界）：只加载边界区域。每个区域在 TECPLOT 中都显示为独立的数据区域。

在组合框下方是 Data File（数据文件）输入框。可以直接在输入框中输入需要加载的数据文件名，或者点击 Select（选择）按钮直接从硬盘上选择数据文件。

14.4 TECPLOT 绘图环境设置

虽然系统缺省设置的绘图环境可以满足大部分绘图的需要，修改绘图环境设置仍然可以提供更多的选择，使所绘制的图形更加丰富多彩。

1. 帧的创建与编辑

对绘图帧的操作主要包括帧的创建与编辑。在缺省设置中，工作区内仅有一个绘图帧。如果需要增加一个绘图帧，可以点击工具栏中的按钮，或者执行下列菜单操作：

Frame -> Create

此时光标将变成十字线形状，在工作区中按住鼠标左键并拖动，就可以画出一个新的绘图帧。

在创建了帧以后，可以对帧进行编辑，包括移动、删除、改变尺寸、修改名称和外观（标题栏、边界和背景）等等。

在进行这些操作之前，首先要选中需要编辑的绘图帧。用鼠标点击帧的标题栏，则在帧的边界上将出现八个黑色方块，表示帧已经被选中，如图 14-19。

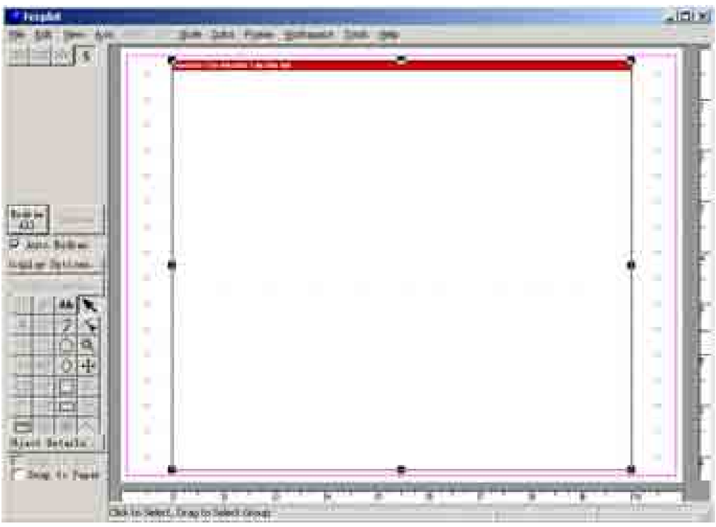


图 14-19 绘图帧处于被选中状态

在帧被选中时，可以进行下列操作：

（1）光标移动到黑色方块上，光标的形状将变成双向箭头，按住鼠标左键并拖动就可以调整帧的大小。

（2）将光标移动到黑色方块以外的区域，则光标形状将变成十字箭头，按住鼠标左键并拖动就可以移动帧的位置。

（3）在选择状态下，按键盘上的 Delete（删除）键，可以删除当前帧。

点击边界以外的区域，或者直接点击鼠标右键可以取消对帧的选择。

双击帧的标题栏可以打开 Edit Current Frame（编辑当前帧）窗口，如图 14-20 所示。

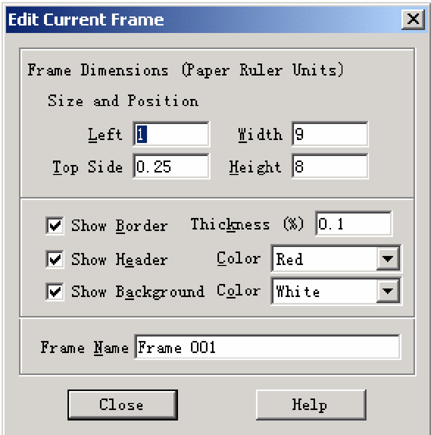


图 14-20 Edit Current Frame（编辑当前帧）窗口

在这个窗口的上部有四个输入框，分别是 Left（帧的左边界位置）、Width（帧的宽度）、Top Side（帧的上边界位置）和 Height（帧的高度）。帧的位置由上边界和左边界的位置确定，实际上是用左上角的坐标值来确定帧的位置。帧的大小则用宽度和高度来定义，因此在四个输入栏中输入相应的数值，就可以精确地定义帧的位置和大小。

在窗口的中部有三行，分别对应于边界（Border）、标题栏（Header）和背景（Background）的设置。每一行的前面都有一个选择框，选中则表示显示相应的图形对象。在缺省设置中，三者均处于被选择状态，表示同时显示边界、标题栏和背景。读者可以尝试取消各选项，看看帧将发生什么样的变化。在右边的输入栏中可以定义边界线的宽度（Thickness），标题栏和背景的颜色（Color）。将三者分别修改为 0.5、Green（绿色）和 Yellow（黄色），并调入安装目录中的示例文件 3dfe.plt，则图形将变成图 14-21 的样子。

在窗口下部为 Frame Name（帧名称），键入新的名称可以改变帧的名称。

利用上述办法创建一个新的帧，并调入示例文件 creatvol.plt，经过位置和大小的调整后，可以得到如图 14-22 所示的图形。

在有多个绘图帧时，可以在 Order Frames（帧的顺序）窗口中调整各个帧的顺序。执行下列菜单操作可以打开这个窗口，如图 14-23 所示：

Frame -> Order Frames...

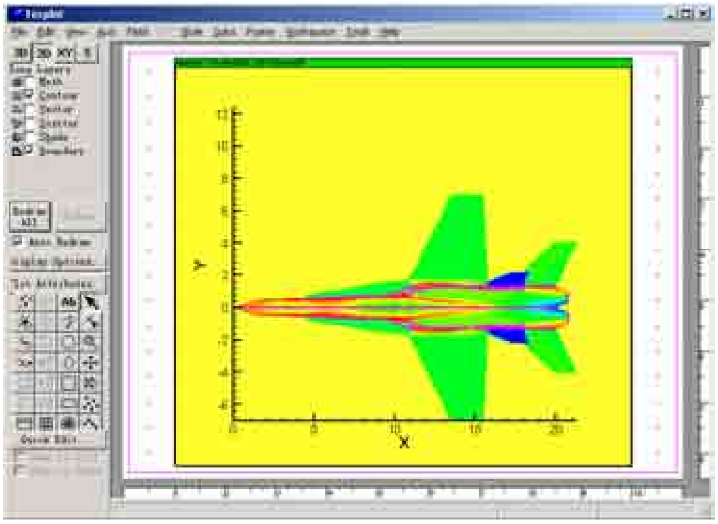


图 14-21 修改帧的边界、标题栏和背景后的图形显示

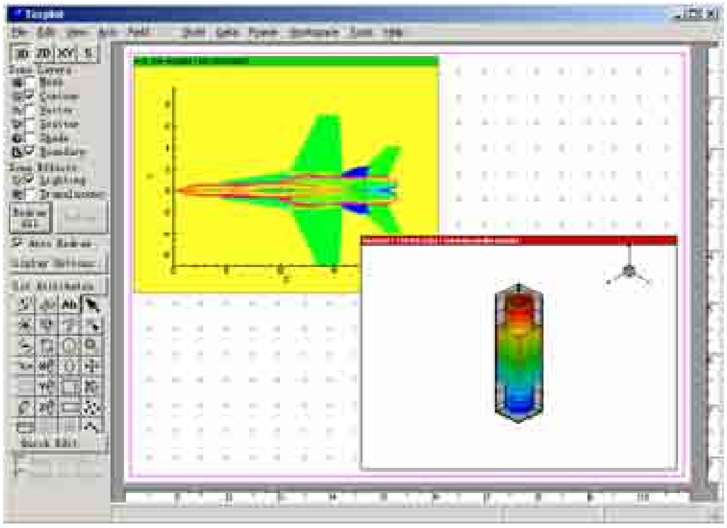


图 14-22 多帧显示



图 14-23 调整帧的显示顺序

点击 Pop（前移）和 Push（后移）按钮，就可以调整当前帧的顺序。

2. 工作区的设置

工作区是进行所有绘图操作的区域。通过对工作区域的正确设置，可以大大方便绘图操作。

（1）纸面控制

在缺省设置中，工作区内的白色背景部分就是所谓的纸面，而绘图帧则放置在纸面以内。纸面的设置主要与打印功能有关——打印机的打印范围就限定在纸面以内。如果仅作屏幕显示之用，绘图帧的位置和大小完全不必受纸面的限制。

纸面的设置在 Paper Setup（纸面设置）窗口中完成，执行下列菜单操作可以弹出这个窗口，如图 14-24 所示：

File -> Paper Setup

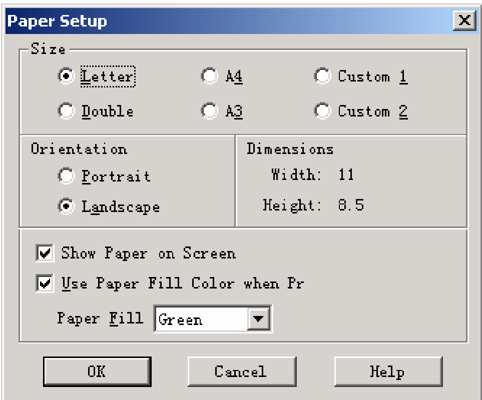


图 14-24 Paper Setup（纸面设置）窗口

在这个窗口中可以设置的项目包括：

1）纸面大小（Size）：其中有六种尺寸可供选择，分别为 Letter（信纸）、A4、A3、Double（大信纸）和 Custom1（自定义尺寸 1）、Custom2（自定义尺寸 2）。纸面的具体尺寸包括 Width（宽度）和 Height（高度），具体数值显示在 Dimensions（尺寸）栏中，采用的长度单位是英寸。

2）纸面方向（Orientation）：即 Portrait（纵向）和 Landscape（横向）。

3）是否显示纸面（Show Paper on Screen）：选中既在屏幕上显示纸面，取消则隐藏纸面。

4）打印时是否按显示打印纸面颜色（Use Paper Fill Color when Print）：选中则打印颜色，取消则不打印。

5）纸面颜色设置（Paper Fill）：可以在旁边的下拉列表中选择一种颜色作为纸面颜色。

（2）网格和标尺的设定。

利用网格可以方便的定位对象，在添加文本和几何图形时可以选择对齐到网格。利用标尺可以方便的放大缩小对象。标尺的显示单位可以选择为厘米（cm）、英寸（in）、点数（pt）或者不显示标尺。要改变网格和标尺设定，需要首先打开 Ruler/Grid（标尺/网格）对话框，其菜单操作步骤如下：

Workspace -> Ruler/Grid

Ruler/Grid（标尺/网格）对话框如图 14-25 所示。

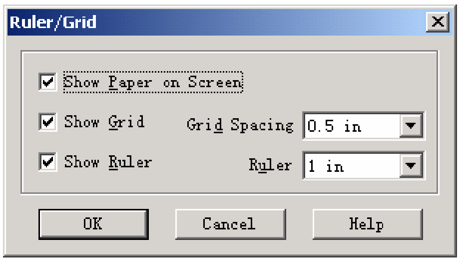


图 14-25 标尺/网格对话框

在这个对话框中，有三个选项，即：

1）是否显示工作纸（Show Paper on Screen）。

2）是否显示网格（Show Grid），并同时设定网格间距（Grid Spacing）。

3）是否显示标尺（Show Ruler），并同时设定标尺刻度单位（Ruler）。

如需要显示相应内容，则在复选框中打勾；如不需选择，则取消打勾。选择结束后，点击 OK 按钮结束操作。

（3）坐标系统

TECPLOT 中包含有多个坐标系统，最为重要的是纸面、帧、二维和三维坐标系统。上述四种坐标系的相互关系及原点位置如图 14-26 所示，其中二维、三维坐标系就是根据数据集合进行绘图的坐标系。二维坐标系又经常被称为网格坐标系。另外，帧坐标系的长宽比例可以任意调节，但是在水平方向和垂直方向的取值范围总是 0 到 100。

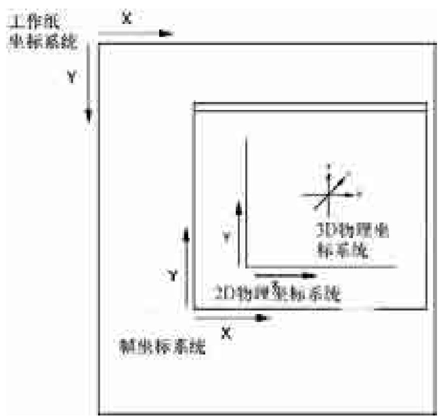


图 14-26 TECPLOT 坐标系统

在绘图过程中，某个对象的大小、坐标是用帧坐标系的长度单位，根据一定的比例定义的。在使用长度单位时可以使用不同的长度单位，例如，如果在帧中插入一段文字，并希望这段文字的起始位置距离帧的左边框 2 英寸，则可以直接在定义文字位置（X 坐标原点）的输入框中输入“2in”。随后系统将根据标尺进行计算，将文字移动到距离帧的左边框 2 英寸处。除了“in（英寸）”，其他可以使用的长度单位还有“pt（点）”、“cm（厘米）”和“pix（象素）”。

3. 视图操作

视图调整可以在 View（视图）菜单和工具栏中完成，主要可以操作的内容包括 Redraw（重画）、Zoom（放大）、Center（居中）、Translate/Magnify（平移/放大）、Probe（取值）等等。

在进行视图操作时，可以用鼠标中键方便地进行放大、缩小操作。操作方法就是在帧内按住鼠标中键，然后上下拖动即可。另外，在帧内按住鼠标右键，可以自由移动图形对象。

点击按钮后，光标形状将变成十字线。然后在图形区域内任意点击一下，就可以弹出 Probe（探测）窗口，如图 14-27 所示。

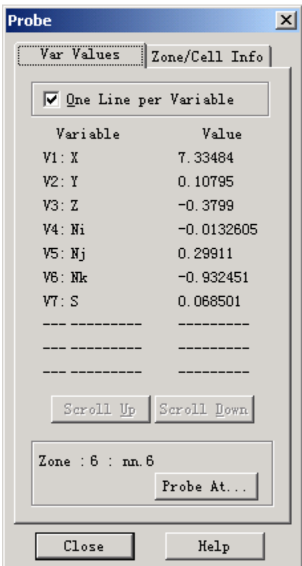


图 14-27 Probe（探测）窗口

窗口中显示的数据就是各变量在被点击点上的数值。在这个窗口下方有一个 Probe At（探测位置）按钮。点击这个按钮可以打开 Probe At（探测位置）窗口，如图 14-28 所示。在这里可以指定被探测点的坐标，再点击下方的 Do Probe（进行探测）按钮，则帧中将以小十字线显示探测点位置，同时在 Probe（探测）窗口中显示该点上的变量值。

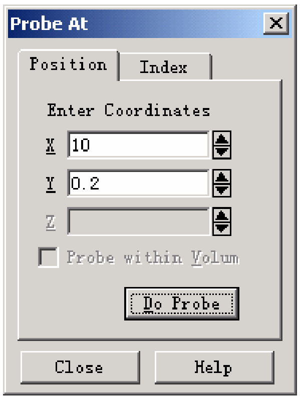


图 14-28 Probe At（探测位置）窗口

在对显示对象进行了一系列操作后，执行下列菜单操作将恢复图像的原始模样：

View -> Fit to Full Size

另外，在 Translate/Magnify（平移/放大）窗口中可以按步骤进行平移和放大、缩小操作，并且可以控制平移距离和放大倍数。执行下列菜单操作可以打开这个窗口，如图 14-29 所示：

View -> Translate/Magnify

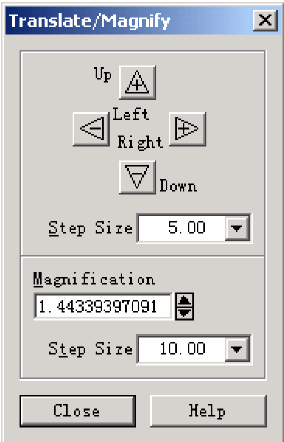


图 14-29 Translate/Magnify（平移/放大）窗口

在这个窗口中点击上部的四个箭头，就可以完成图形对象向上、下、左、右的移动操作。点击 Magnification（放大）输入栏右边的箭头，可以完成图形对象的放大和缩小操作，其中点击向上箭头为放大，点击向下箭头为缩小。

执行下列菜单操作则恢复到图形在最后一次操作时的状态：

View -> Last

除上述操作外，还可以对帧进行剪切（Cut）、复制（Copy）、粘贴（Post）、清除（Clear）等操作。这些操作均可以在 Edit（编辑）菜单中完成，其方法与其他 WINDOWS 风格的软

件相同，这里就不再一一赘述。

本章简单介绍了 TECPLOT 的基本情况和常用操作，下一章将介绍后处理中常用的几种图形显示方法。

第十五章 TECPLOT 实战

本章通过一些具体实例讲解 TECPLOT 的使用方法，例子中所有的数据文件都使用 TECPLOT 自带的数据文件。

15.1 绘制 XY 曲线

TECPLOT 中的所有 XY 曲线都是由一个或者多个 XY 数据对构成的。XY 数据对及它们之间的函数关系，在 TECPLOT 中被称为 XY 图形。XY 图形有三种形式：

（1）折线图（Lines）：用线段连接所有的数据点。

（2）符号图（Symbols）：每个数据点由一个符号代表，例如圆、三角形或方形等等。

（3）柱状图（Bars）：每个数据点由一个水平或垂直柱代表。

用于绘制 XY 曲线的数据可以直接在 TECPLOT 中输入，也可以加载硬盘上的数据文件用于绘制曲线。

1. 直接输入 XY 数据

XY 数据的输入在菜单项 Data（数据）下进行，顺序点击：

Data -> Create Zone -> Enter XY Values

将弹出一个名为 Enter XY Value to Create a Zone（输入 XY 数据以创建区域）的窗口，如图 15-1 所示。

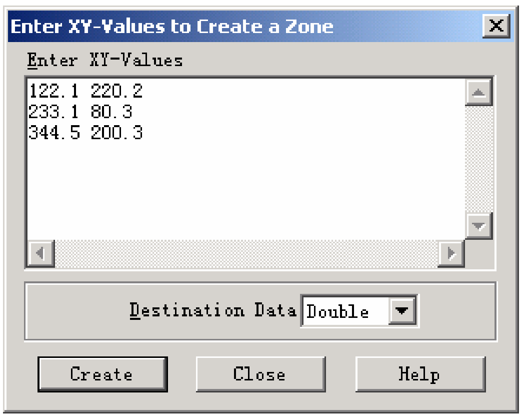


图 15-1 直接输入 XY 数据

首先在上方的窗口中输入二列数据，其中第一列对应于 X 变量，第二列对应于 Y 变量；其次在 Destination Data（目标数据）中指定数据类型；最后点击 Create（创建）按钮，就完成 XY 曲线绘制。

2. 载入数据文件

顺序点击菜单：

File -> Load Data File（s）

将打开一个提示信息窗口，要求用户在 3 个选项中进行选择。这 3 个选项是：

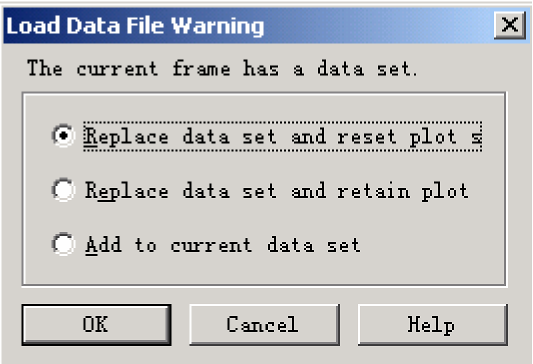


图 15-2 信息提示窗口

（1）用新的数据代替原数据，并重置图形设置。

（2）用新的数据代替原数据，并保持原图形设置。

（3）将新的数据加入现有数据集合。

根据需要进行选择后，点击 OK 按钮就可以打开文件选择窗口。载入所需的数据文件，就完成数据文件载入过程。图形窗口中将按缺省方式显示对应的 XY 曲线。

3. 曲线的调整

下面我们结合 TECPLOT 自带的 rain.plt 文件进行实例讲解 XY 曲线的调整。

（1）打开 rain.plt 文件。

1）点击菜单 File，选择 New Layout。

2）在 TECPLOT 安装目录 TEC90 下，选择子目录 Demo/plt/，并载入文件 rain.plt。

（2）调整绘图选项

1）在 XY 菜单中，选择 Define XY Mappings（定义 XY 图形）项。在按住 shift 键的同时，点击并选中 Map Num（图形编号）下面的 2 和 3，然后点击 Map Show（图形显示）按钮会出现一个选择窗口（如图 15-3 所示），选择 Activate（激活），则将同时显示编号为 1、2 和 3 的三条曲线。

2）在 Define XY-Mappings（定义 XY 图形）窗口中还有几个标签，可以分别用来改变

曲线形式，定义 XY 轴变量，改变柱状图形式等等。由于其操作过程非常简单，这里不再一一赘述。读者可以自己尝试改变各种设置，并观察所产生的效果。图 15-4 为改变设置后的一个示例图。

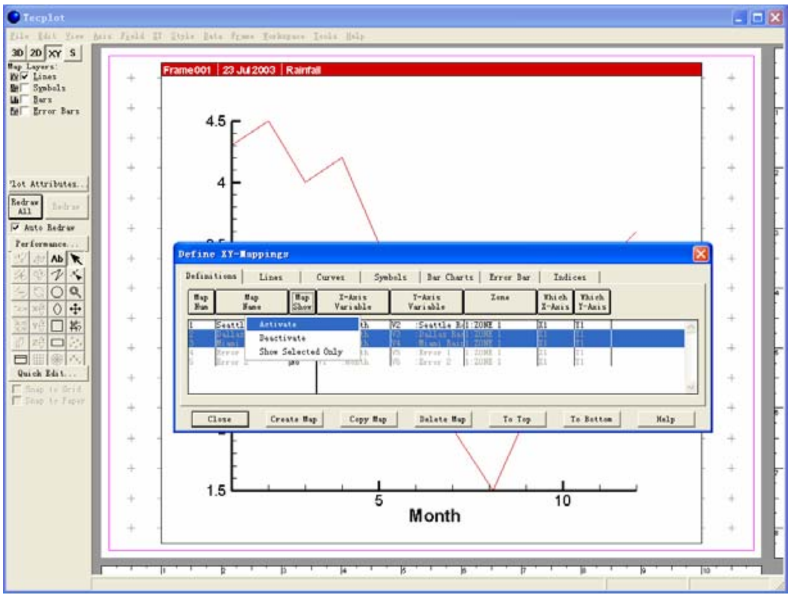


图 15-3 显示 XY 曲线

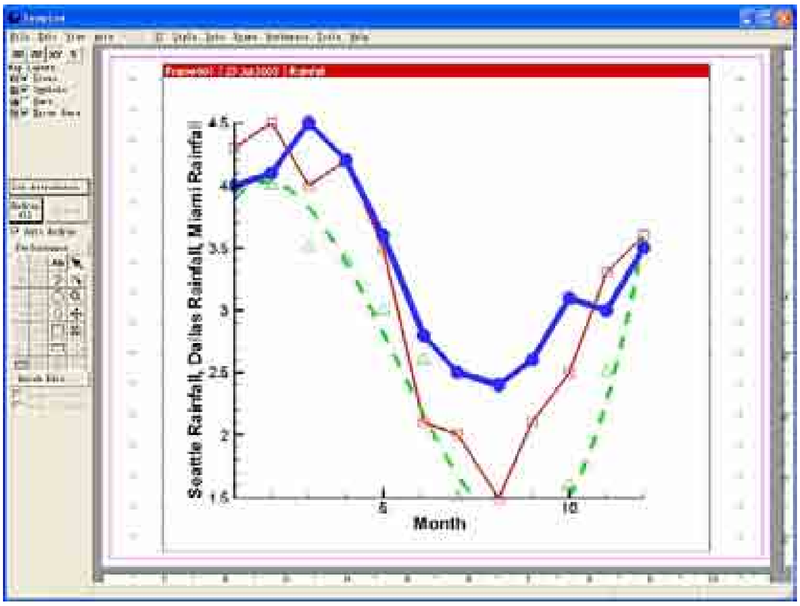


图 15-4 XY 曲线绘制示例

15.2 绘制矢量图

矢量图用于表现速度、作用力等矢量的大小和方向。矢量图分二维矢量和三维矢量，分别对应于二维流场计算和三维流场计算。在读入数据文件后，点击左上方工具栏中的 3D 或 2D 按钮，在按钮下方就可以发现 Vector（矢量）选项。点击 Vector（矢量）选项，则可以自动弹出 Select Variables（选择变量）窗口，如图 15-5 所示。

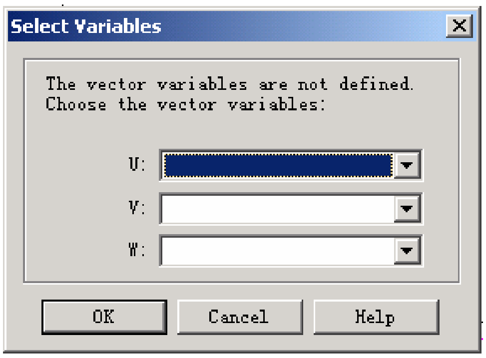


图 15-5 确定矢量分量

图 15-5 为三维矢量情况下的窗口形式，在二维情况下则仅有 U、V 两个分量。在这个窗口中设定矢量分量所对应的数据。设置完毕后，点击 OK 按钮，则帧中将自动显示在缺省设置下的矢量图。

矢量图的显示方式也是可以进行调整的。下面利用 TECPLOT 自带的示例文件 velocity.plt 对调整方法进行说明。Velocity.plt 文件位于 TECPLOT 的安装目录 TEC90 下。顺序进入 Demo/plt 子目录，最后选择 velocity.plt 文件完成载入过程。

下面对矢量图进行绘制属性设定：

（1）在工具栏中，取消对 Mesh（网格）的选择，并选择 Vector（矢量），则原来绘图区中的网格消失，并弹出 Select Variables（选择变量）窗口。在 U 右面的下拉列表中选择 V4：U/RFC，V 右面的下拉列表中选择 V5：V/RFC，点击 OK 按钮完成分量设置。

（2）由于默认的矢量箭头颜色为红色，不能直观反映出速度的大小，所以下面选择新的矢量着色。点击工具栏中的 Plot Attributes（绘图属性）按钮，打开绘图属性窗口，如图 15-6 所示。

在 Vector（矢量）标签下，点击 Line color（线颜色），在随即弹出的颜色选择窗口中选择 MultiColor（多色），或其它单色，并确定颜色将随哪个变量的值发生变化。最后调整矢量宽度为 0.3%，所得图形如图 15-7 所示。

其它可以设置的参数包括矢量箭头的起点位置、长度、线型，箭头的形式（空心、实心或普通），是否激活矢量区，编辑矢量区的名称等等。建议读者自己进行尝试，这里限于篇幅就不再赘述。

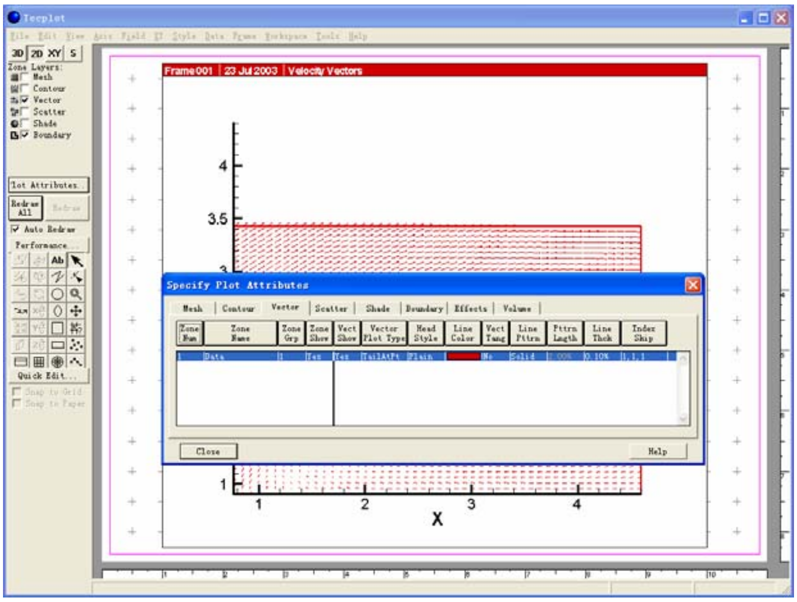


图 15-6 绘图属性窗口。

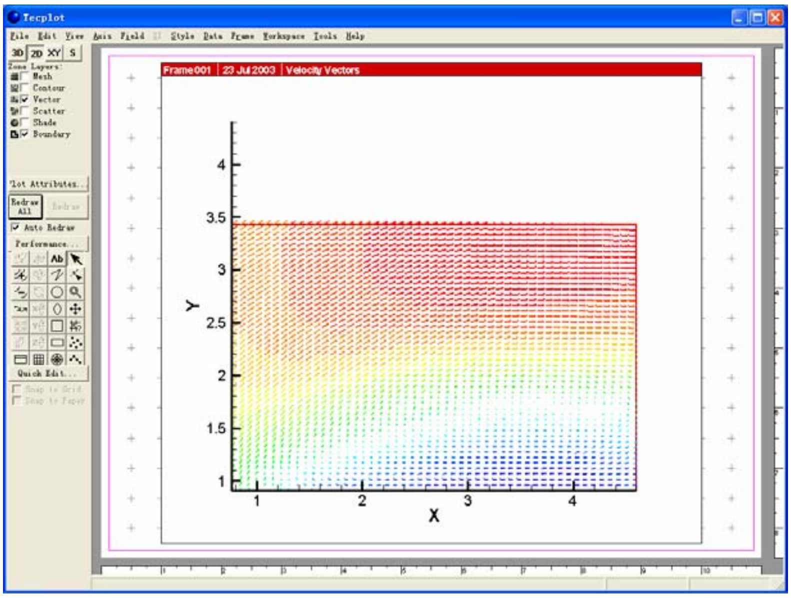


图 15-7 等值线颜色着色矢量图

除了绘制矢量线外，还可以在矢量图上绘制流动迹线。在工具栏中选择，然后用鼠标在绘图区中单点或上下拖动鼠标，便可以绘制出迹线图，如图 15-8 所示。

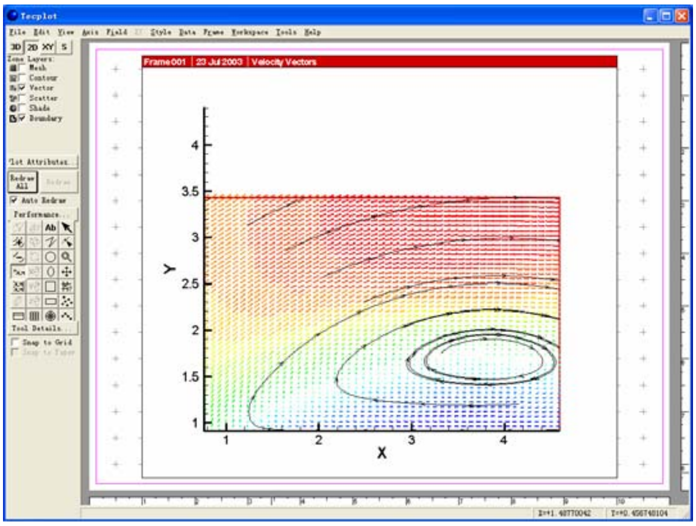


图 15-8 迹线图

15.3 绘制等值线图

等值线是观察流场变量变化的常用工具。绘制等值线一般包括五个步骤：

（1）加载数据文件。

（2）点击工具栏中的 Contour（等值线）选项，激活等值线图层。

（3）选择目标变量，绘制等值线。

（4）调整绘图参数，最终得到理想的等值线图。

（5）保存图形文件。

在上述五个步骤中，第（4）步比较复杂，这里用一个例子稍微详细地进行讲解。为了保证叙述的连贯性，这里依然使用文件 velocity.plt 讲解等值线参数的调整：

（1）首先，如上一节那样，将 velocity.plt 加载到系统中来。

（2）在工具栏中去掉 mesh 前的选择标记，并选择 Contour（等值线），在弹出的 ContourVariable（等值线变量）窗口的选择框中选择 R/RFR，然后点击 Close 按钮关闭选择框，就可以得到如图 15-9 所示的变量 R/RFR 的等值线图。需要提醒读者注意的是，在选定等值线变量后，还可以重新启动 ContourVariable（等值线变量）窗口，重新选择等值线变量，其菜单操作顺序为：

Field -> Contour -> Contour Variables

（3）下面调整等值线的密度和数值。

1）进行下列菜单操作将打开 Contour Levels（等值线层级）窗口，如图 15-10 所示：

Field -> Contour -> Contour Levels

2）首先重新设定总的级数，点击 Reset Levels（重置层级），并在弹出的输入栏中输入 10。

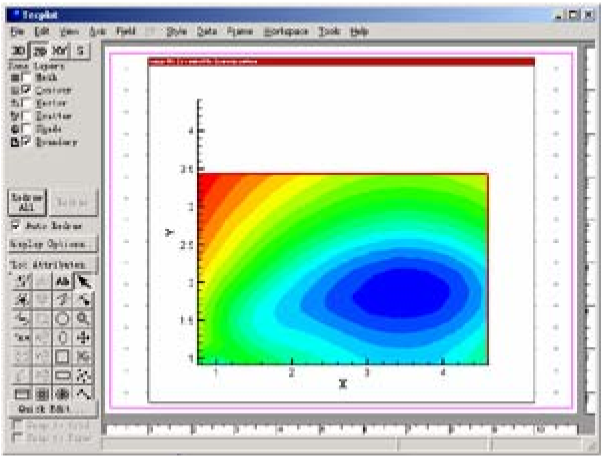


图 15-9 等值线图

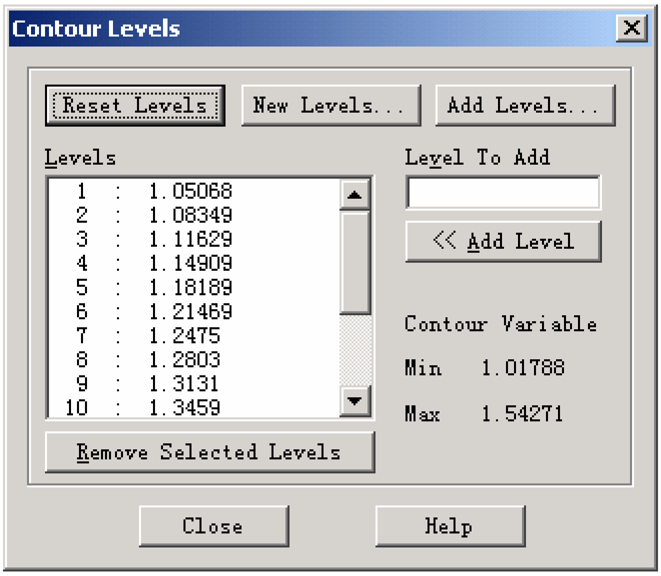


图 15-10 等值线层级窗口

3）然后调整等级密度，点击 New Levels（新层级），保持 Min，Max and Numberof Levels（最小值、最大值和级数）被选中，然后在 Minimum Level（最小层级）中输入 1.0，在 Maximum Level（最大层级）中输入 1.5，如图 15-11 所示。在这里也可以在 Number of Levels（级数）中直接输入级数，进行等值线级数的调整。

除了 Min,Max and Number of Levels（最小值、最大值和级数）选项外，还可以选择 Min,Max and Delta（最小值、最大值和级间增量）和 Exponential Distribution（指数分布）。从三者的名称中既可看出：Min,Max and Number of Levels（最小值、最大值和级数）是通过指定场变量的最小值、最大值和级数来定义等值线的；Min,Max and Delta（最小值、最大值和级间增量）是通过指定场变量的最小、最大值和级与级之间的数值增量来定义等值

线的；而 Exponential Distribution（指数分布）则用指数分布形式绘制等值线。

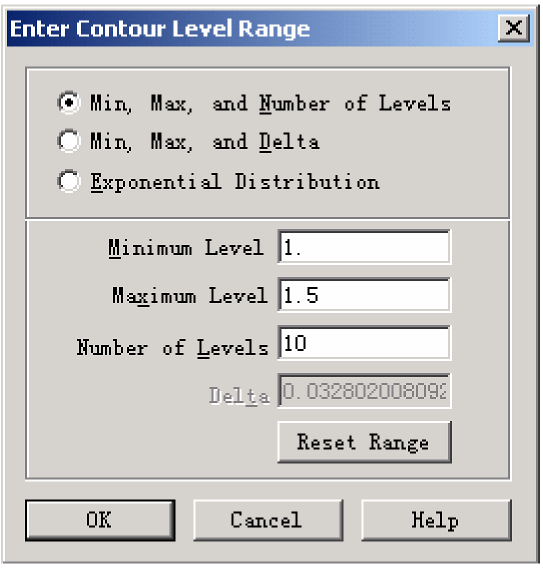


图 15-11 调整等值线密度

4）下面加上等值线的轮廓线。点击工具栏的 Plot Attribute（绘图属性）按钮，在弹出的对话框中单击 Contour Plot Type（等值线绘图类型），选择 Both Lines & Flood（画线并填充），等值线图形中便有了轮廓线，可以调整轮廓线的线型、颜色和宽度等参数，以达到最佳的显示效果。生成的图形应如图 15-12 所示：

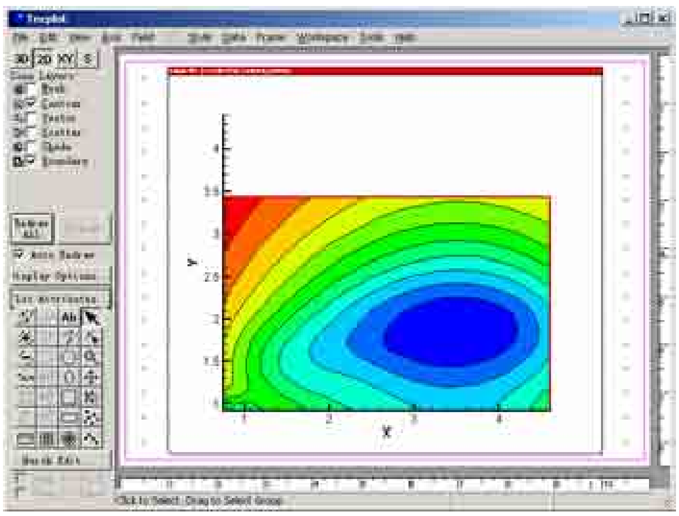


图 15-12 调整后的等值线及其轮廓线。

5）最后加上图例并在轮廓线上标值。执行下列菜单操作，弹出窗口如图 15-13 所示：

Field -> Contour -> Contour Legend

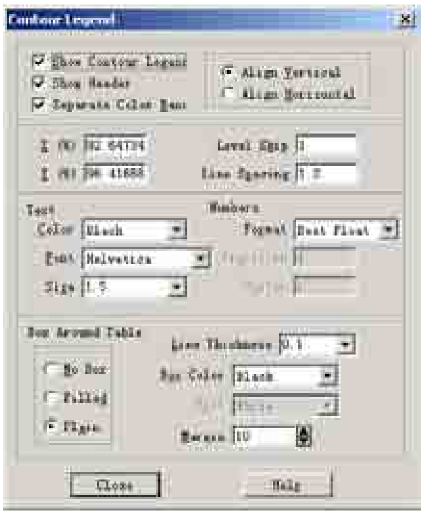
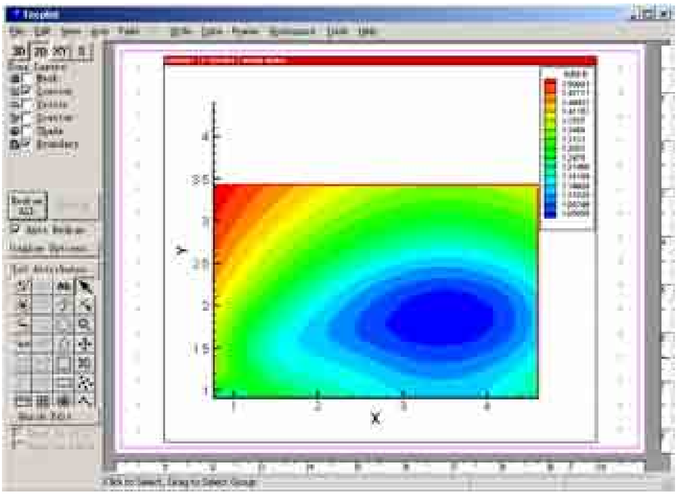


图 15-13 图例窗口

选中 Show Contour Legend（显示等值线图例）；选择 Align Vertical（垂直对齐）；在 Text 组合框中调整文本颜色（Color）、字体（Font）、尺寸（Size）和格式（Format）；通过 Box Around Table（边框设置）决定是否为图例加边框，并确定边框的颜色（Box Color）、线型宽度（Line Thickness）和间距（Margin）。调整后的窗口应如图 15-14 所示。


图 15-14 等值线及其图例

点击工具栏上的按钮，然后在希望标注的地方单击鼠标左键，便可以得到当地流场变量的数值，完成对等值线的数字标注。如果需要对标注方式进行修改，可以在 Contour Labels（等值线标注）窗口中进行。启动这个窗口的菜单操作顺序为：

Field -> Contour -> Contour Labels

在这个窗口中，除了有设置标注字体、格式、颜色的选项外，还有两个常用选项，一个是 Generate Automatic Labels（自动创建标注），选中这个选项，则等值线上将自动出现数值标注；另一个是 Align Next User Position（与下一个用户位置对齐），选中这个选项，则等值线的标注将全部与等值线方向保持一致，如果取消这个选项，则数值标注将保持水平。最后一个要介绍的是 Clear All Contour Labels（清除所有等值线标注）按钮，点击这个按钮，并在弹出的消息框中点击 OK 按钮确认后，则所有标注将被清除。图 15-15 是人工加的标注，其中取消了对 Align Next User Position（与下一个用户位置对齐）的选择，因此所有标注均保持水平。

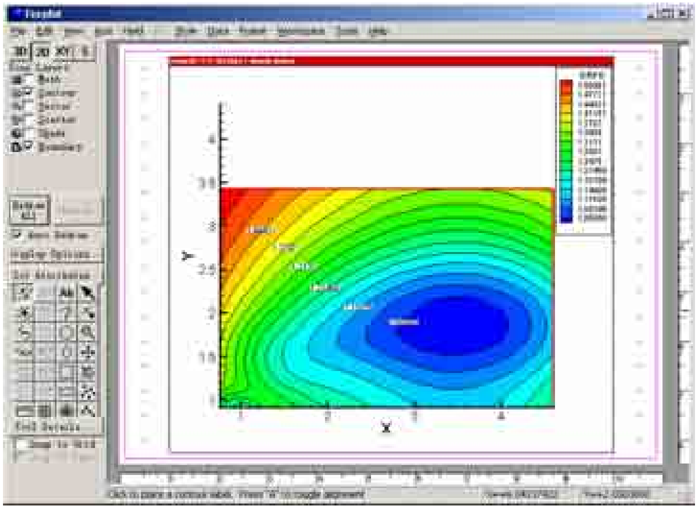


图 15-15 等值线及其数值标注

15.4 绘制流线图

在绘制矢量图的过程中，已经绘制了一个简单的二维流线图。本节中我们将绘制更为复杂的三维体流线图。

（1）首先打开 fetetra2.plt 文件。该文件位于 TECPLOT 的安装目录 TEC90 下，路径为 TEC90/Demo/plt/fetetra2.plt。

（2）在工具栏左上方的 Zone Layers（区域层）部分，去掉 Mesh 前的☑，选中 Contour 前的☑。在弹出的 Contour variable（等值线变量）对话框中选择 V4：U（ms），关闭对话框。

（3）在工具栏中的 Zone Effects（区域效果）部分，选中 Translucency（透明），这样 TECPLOT 会对整个绘图进行透明化处理，透明程度的设定可以在工具栏中 Plot Attribute（绘图属性）中的 Effect 选项进行设定。

完成上面操作后，整个 TECPLOT 界面应如图 15-16。

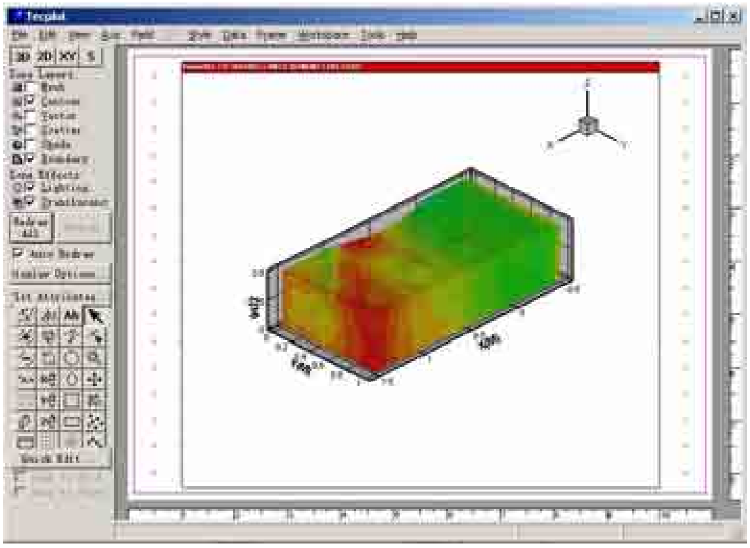


图 15-16 等值线图

（4）添加流线。首先设定流线属性。执行下列菜单操作：

Field -> Streamtrace Placement

因为前面没有设定矢量的三个分量，所以这里首先打开的是 Select Variable（选择变量）窗口。在这个窗口中指定矢量的三个分量分别为 V4：U（m/s）、V5：V（m/s）和 V6：W（m/s），点击 Close 按钮关闭窗口。随即将自动出现 Streamtrace Placement（流线位移）窗口。在这个窗口中设定 Format（格式）为 Volume Rod。Direction（方向）为 Both（两者都有）。然后点击工具栏中的，在图中合适位置添加流线，如图 15-14。

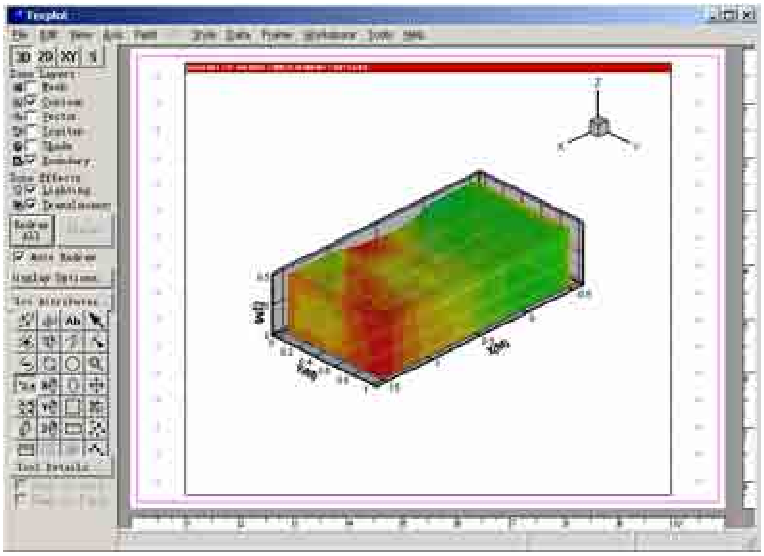


图 15-17 添加流线后的窗口。

由于流线默认为白色，且线宽太小，所以要作进一步调整。执行下列菜单操作，打开 Streamtrace Details（流线细节）窗口，如图 15-18 所示：

Field -> Streamtrace Details

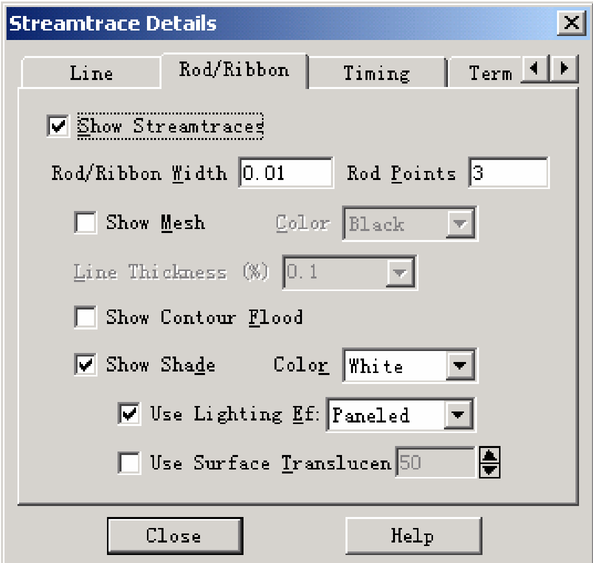


图 15-18 流线细节窗口

在窗口中调整 Rod/Ribbon Width（杆或条带的宽度）值为 0.03，选择 color（颜色）为 Blue（蓝色）。调整后的流线如图 15-19 所示。

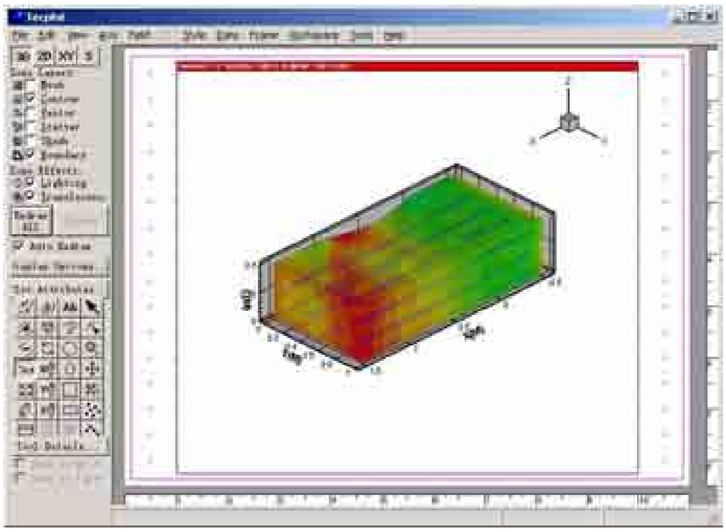


图 15-19 调整后的流线图。

15.5 绘制散点图

所谓的散点图便是用一系列位于数值点位置上的符号来表示数据的一种图形。这些代表数值的符号可以根据变量值的大小而有大小或者颜色的不同，当然也可以是相同大小或者相同颜色。与等值线图相比散点图最大的优势便在于其不需要网格结构，可以对不规则的数据进行处理。图 15-20 便是一个散点图的示例。

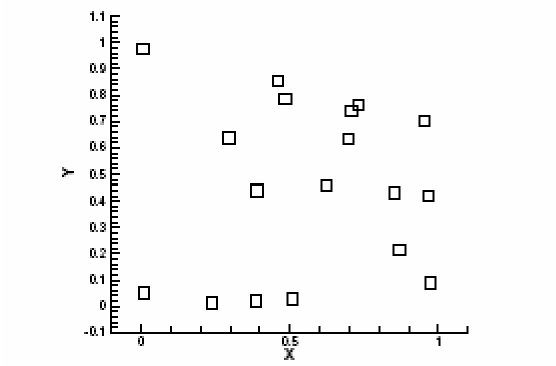


图 15-20 散点图示例。

散点图的绘制相对简单，但要想得到很好的标识数据的散点图要需要进行大量的调整工作。下面通过一个例子来讲解调整方法：

（1）首先打开文件 sylinder.plt。该文件位于 TECPLOT 的安装目录 TEC90 下，路径为 Demo/plt/cylinder.plt。

（2）数据载入后去掉对 Mesh（网格）的选择，并选择 Scatter（散点图），结果如图 15-21 所示。

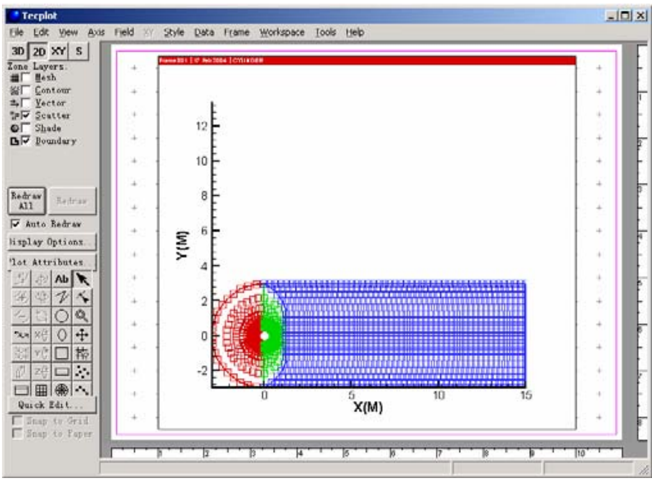


图 15-21 散点图

点击工具栏中的 Plot Attribute（绘图属性）按钮，并打开 Scatter（散点图）标签。散点图的调整主要就是在这里完成的，如图 15-22 所示。

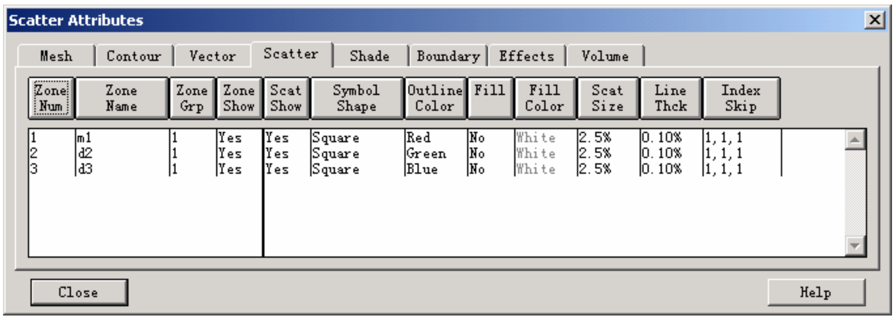


图 15-22 散点图调整窗口

由于图 15-21 中的点太密，根本无法观察，所以首先调整数据密度。首先选择三个数据区，然后点击窗口右端的 Index Skip，选择 Enter Skip，并将 I、J 二参数的值都修改为 3。

在窗口中央的 Symbol Shape（符号形状）中可以修改各数据区的显示符号。假定 1 区采用默认方形（square），2 区采用圆（circle），3 区则采用菱形（diamond）。

最后在 Fill（填充）下选择 Yes，表示需要采用填充形式。然后在 Fill Color（填充色）中分别调整三个数据区填充色，在 Outline Color（轮廓色）中调整轮廓色。调整后的散点图如图 15-20 所示。

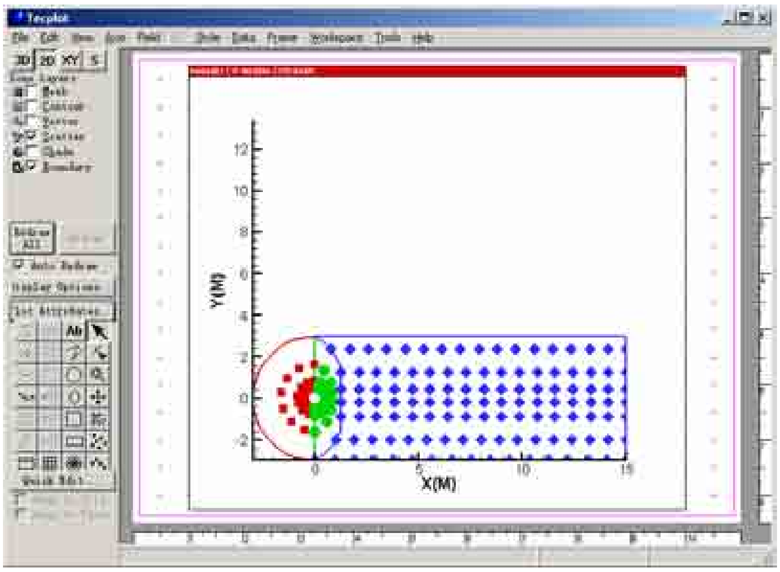


图 15-20 调整后的散点图

15.6 绘制三维流场剖面图

三维流场图（矢量图、散点图、流线图等）的处理方法和二维数据处理方法基本相同。TECPLOT 中还有针对三维数据的特殊绘图格式——剖面图。剖面图可以用来观察流场内部数据变化，所以也是经常使用的后处理工具。剖面图分三种类型：第一种是根据数值大小进行的剖切，称为数值剖切（Value-Blanking）；第二种是根据有序数据在 X、Y、Z 方向上的序列号 IJK 的取值范围进行的剖切，称为 IJK 剖切（IJK-Blanking）；第三种是根据图形到屏幕之间的距离进行的剖切，称为深度剖切（Depth-Blanking）。

剖面图的制作是在 Style（风格）菜单中进行的。这里以 TECPLOT 提供的示例文件 ijkortho.plt 为例逐个进行讲解。示例文件 ijkortho.plt 位于 TECPLOT 的安装目录 TEC90 下，路径为 Demo/plt/ijkortho.plt。首先加载 ijkortho.plt 文件，然后取消对 Mesh（网格）的选择，并选择 Contour（等值线），然后将 V5：E 设为显示变量，结果如图 15-21 所示。

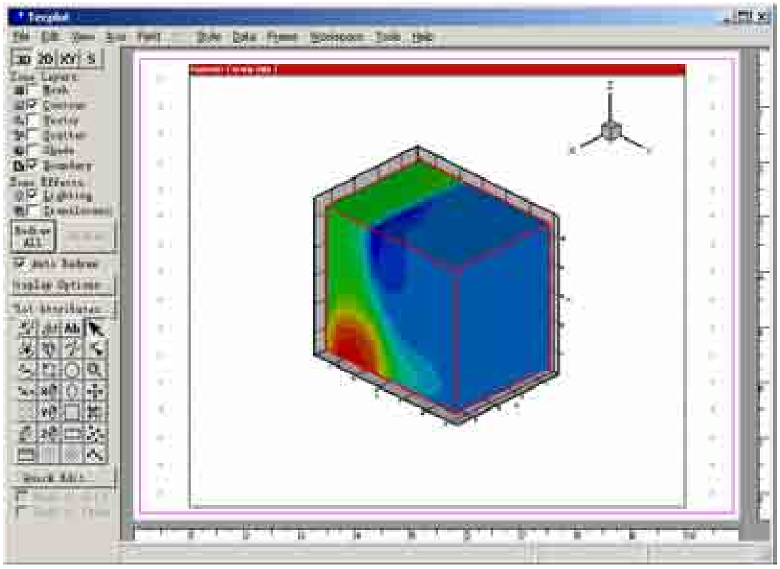


图 15-21 示例文件 ijkortho.plt 的等值线图

1. 数值剖切（Value-Blanking）

数值剖切将剖切范围与某个变量相联系，根据变量的变化范围确定剖切区域。数值剖切的设置是在 Value-Blanking（数值剖切）窗口中进行的。执行下列菜单操作，打开这个窗口，如图 15-22 所示：

Style -> Value Blanking

首先，选中 Include Value Blanking（包含数值剖切）选项，表示在图形显示中将使用数值剖切。



图 15-22 数值剖切窗口

下面的 Blank entire cell when（在满足下列条件时即去除网格单元）选项，表示在何种情况下去除整个单元。可供选择的选项包括：

（1）all corners are blanked，即仅在网格单元的所有角点位于删除范围内的话，才删除整个网格单元。

（2）any corner is blanked，即如果网格单元的任意一个角位于删除范围内的话，则删除整个网格单元。

（3）primary corner is blanked，即在主角点位于删除范围时，则删除整个网格单元。主角点指网格单元中，指标序列号最小的角点。

这里我们可以保持系统缺省设置不变。

下面由 1 到 8 排号的八个按钮，分别代表八个约束条件。首先按下 1 号按钮，并选中下面的 Active（激活）选项。然后在 Blank when（删除条件）组合框中设置删除条件。

删除条件的第一栏为一个变量列表，其中列出了所有可供选择的变量名。这里保持系统缺省选择，以 X 为控制变量。

第二栏为条件列表，其中包括 is less than or equal to（小于等于）和 is greater than or equal to（大于等于）。这里选择 is greater than or equal to（大于等于）。

再下面是二个选项，一个是 Variable（变量），另一个是 Constant（常数）。如果选择 Variable（变量），则下面的输入框变为变量列表。这里选择 Y，则整个删除条件的含义为——当 X>Y 时，删除整个网格单元。结果如图 15-23 所示。

如果选择 Constant（常数），则在下面的输入框中输入一个常数值，比如 6，则整个结果显示如图 15-24 所示。

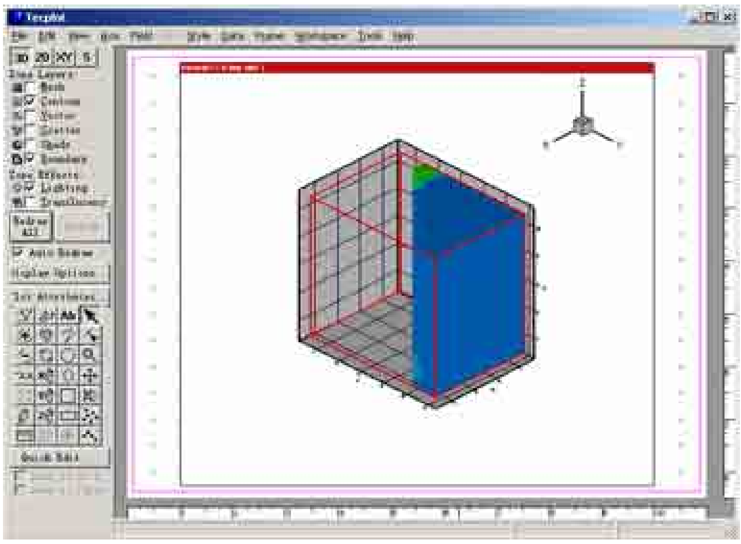


图 15-23 X>Y 时的剖面图

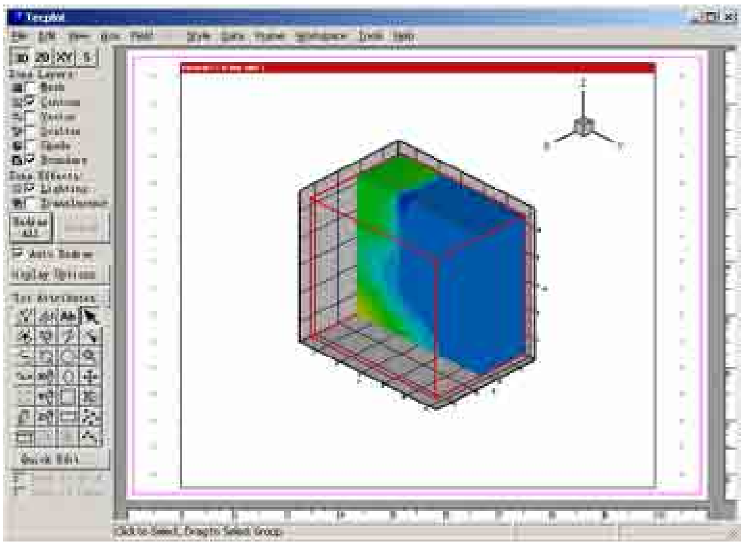


图 15-24 X>6 时的剖面图

在实际绘图中，可能需要同时采用多个约束条件。只要按照上面介绍的方法，逐个设定这些约束条件，即可获得不同形式的数值剖切剖面图。

2. IJK 剖切（IJK-Blanking）

IJK 剖切是根据 IJK 有序数据的序号范围，进行剖切的一种形式。IJK 剖切的设置在 IJK Blanking（IJK 剖切）窗口中进行。执行下列菜单操作打开这个窗口，如图 15-25 所示：

Style -> IJK Blanking

与数值剖切类似，首先选中 Include IJK Blanking（包含 IJK 剖切）选项，表示将在绘图

中使用 IJK 剖切。

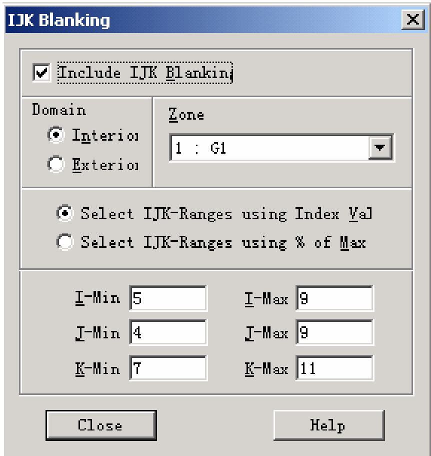


图 15-25 IJK 剖切窗口

在 Domain（域）组合框中有二个选项，一个是 Interior（内部），意思是在下面数据范围以内的网格单元将被删除；一个是 Exterior（外部），意思是在下面数据范围以外的网格单元将被删除。

保持 Zone（数据区）的缺省设置不变。下面的选项中，一个是 Select IJK-Ranges using Index Value，即根据有序数据的序号范围执行删除操作；另一个是 Select IJK-Ranges using % of Max，即根据数据最大值的百分比范围执行删除操作。这里选择第一项，即根据数据的序号执行删除操作。

在最下面的数据范围输入框中，有三行数据，分别对应于 I、J 和 K 三个方向的序号范围。这里将其分别设置为 I：5—9，J：4—9，K：7—11，结果显示如图 15-26 所示。

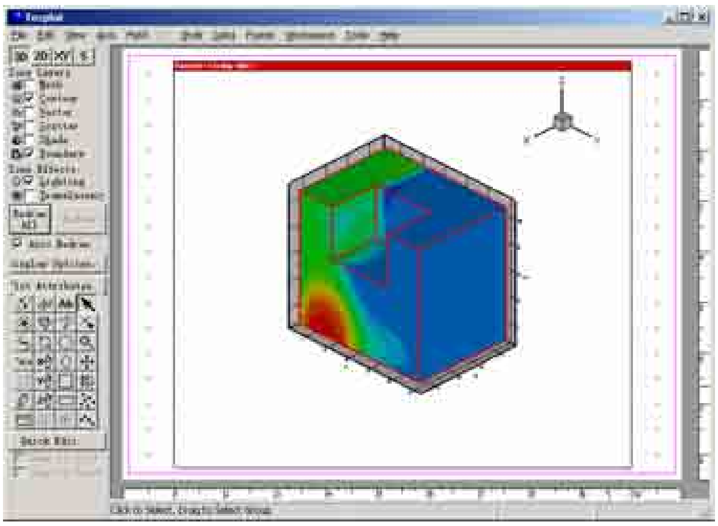


图 15-26 IJK 剖切示例

3. 深度剖切（Depth-Blanking）

深度剖切是以屏幕所在平面为基准，以图形距离屏幕的距离为控制参数进行的剖切。深度剖切在 Depth Blanking（深度剖切）窗口中进行设置。执行下列菜单操作，可以打开这个窗口，如图 15-27 所示：

Style -> 3D Depth Blanking

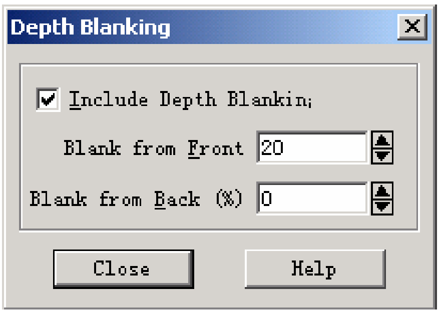


图 15-27 深度剖切窗口

这个窗口中有二个输入栏，第一栏为 Blank from Front（从前面剖切），第二栏为 Blank from Back（%）（从后面剖切）。在第一栏中输入距离 20，则显示结果如图 15-28 所示。

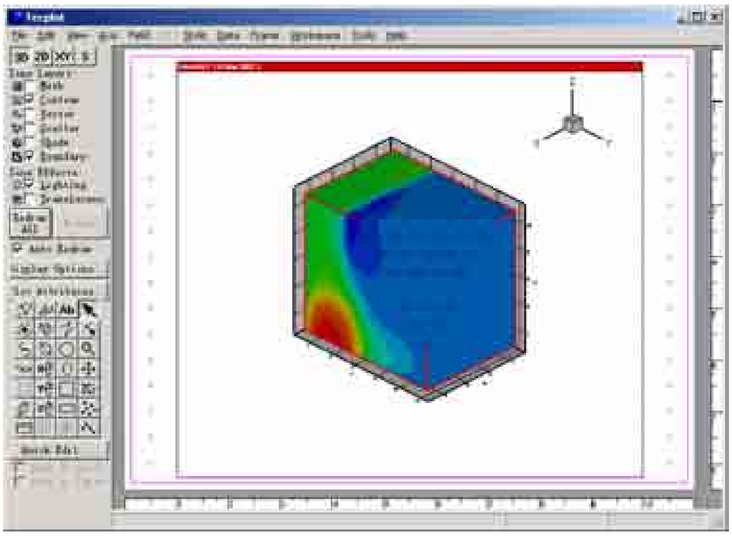


图 15-28 从前面剖切后的结果

从图 15-28 中可见，靠近屏幕的角已经被切除。

4. 围墙图

最后介绍的一种绘图方式，通常被称为围墙图。之所以称其为围墙图，是因为它的形状类似于用围墙把平面分割成许多小格子。

围墙图的特别之处在于要绘出底面数据，然后绘出垂直于底面的数据作为围墙。用这种处理方式可以用切片方式观察流场内变量的变化。其绘制方法如下：

（1）点击工具栏中 Plot Attribute（绘图属性）按钮。

（2）点击进入 Volume（体积）标签，如图 15-29 所示。

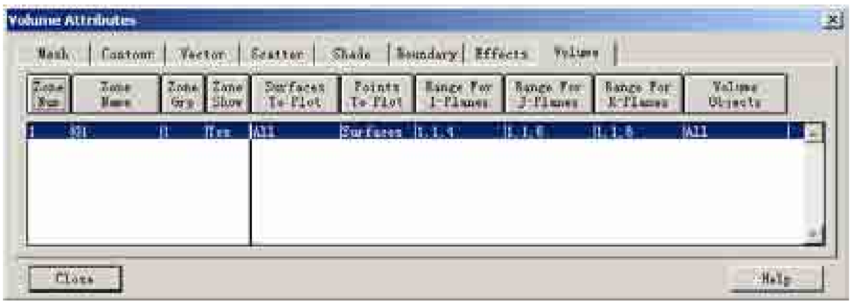


图 15-29 Volume 标签中的可选项

（3）在 Surfaces To Plot（绘制表面）下选择 I- & J- & K-planes（I、J、K 平面）。窗口应如图 15-30 所示。

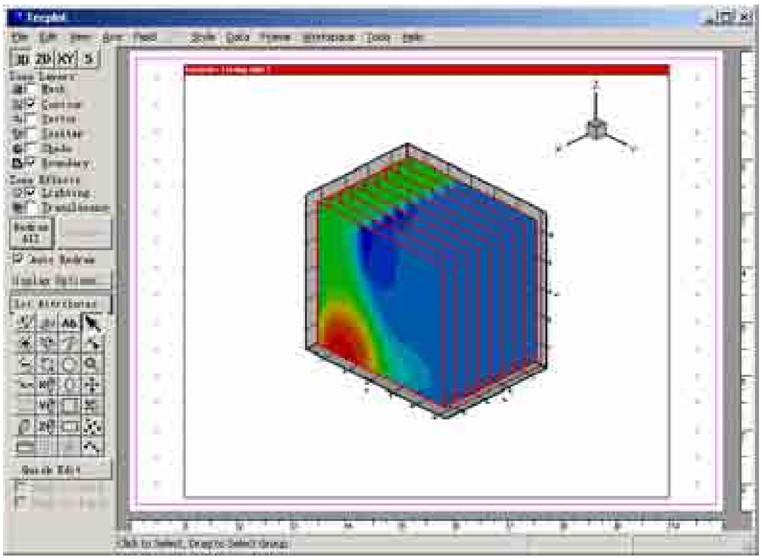


图 15-30 围墙图

可以看到只有沿 X 方向有了围墙。为了在 Y 和 Z 方向也产生围墙。需要改变其范围为最大（mx）。在 Range For I-Planes（I 平面范围），Range For J-Planes（J 平面范围）和 Range For k-Planes（K 平面范围）中分别调整数值为：

I：Begin（起始）：1；End（终止）：mx；Skip（跳过）：mx；

J：Begin（起始）：1；End（终止）：mx；Skip（跳过）：5；

K：Begin（起始）：1；End（终止）：mx；Skip（跳过）：6；

生成的图形如图 15-31 所示。

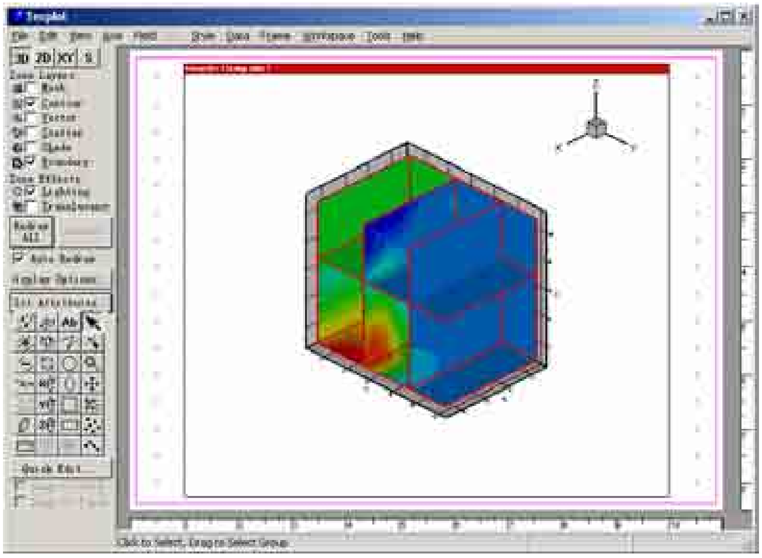


图 15-31 围墙图

本章介绍了应用 TECPLOT 可以绘制的几种图形。这些图形在后处理过程中，都是经常用到的。读者根据本章的讲解就可以基本掌握其绘制方法，绘制出漂亮的后处理图形。

第五篇　高级话题

第 16 章 场函数定义

了解场变量是利用FLUENT进行工作的基础。本章第16.1节和16.2节综合介绍了与场变量相关的基础知识。在第16.3节里，表16.3.1-16.3.13以分类清单的形式列举出了各变量。第16.4节介绍了按照字母顺序排列的场变量清单及定义。本节涵盖了可能出现在许多面板中的变量。第16.5节介绍了如何计算自定义的场函数。

16.1：节点和单元的值

16.2：速度报告选项

16.3：场变量分类清单

16.4：按字母顺序排列的场变量清单及定义

16.5：调整场函数

16.1　节点和单元的值

场变量值可以分为节点值和单元值。节点值是定义在网格节点上的变量的值，单元值则是定义在网格单元上的值。

在后面的论述中，“表面（surface）”是指可以在Surface菜单中建立和处理的面、线或点的集合。在大多数情况下，这些表面是网格单元面或其他表面与流场变量等值面相交形成的。

16.1.1　单元值

FLUENT把大部分的变量设为单元变量，单元变量的值代表的是网格单元上变量的平均值。表面单元值是与其它面、线存在相交关系的单元上的变量值。由于面和线是等值面或等值线与网格单元相交形成的，所以表面单元值是唯一确定的。在边界上，单元值等于邻近边界的单元上的变量值。

16.1.2　节点值

节点值被显式定义为单元变量的平均值。边界条件定义了场变量在边界上的值，所以在边界区域上节点值是被显式定义的。另外，有些变量（例如，节点的坐标）的值可以在节点上直接定义。对于大多数变量，节点值则只能取为节点周围所有网格单元上数据的平均值。

计算节点的值包括两个步骤：

（1）初始化时，所有节点值取为节点周围单元值的平均值。

（2）在边界上，节点值等于边界值。

例如，在图16-1中，节点n1的值等于它周围单元（c1到c6）上数据的平均值。节点n2是边界上的一个节点，如果在计算中变量在边界上有确切的值，那么这个值就是边界节点n2上相应变量的值，而不是单元c1、c6、c7的平均值。

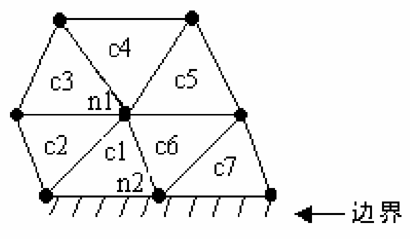


图 16-1 计算节点值

需要注意的是，自定义的场函数无法用显式边界条件中的数值加以定义。

表面上的节点值是通过对网格节点数据进行线性插值获得的。在域表面上，表面的节点就是区域的节点，因此节点值也是相同的。对于等值面和等值线，其节点值是用与等值面相交的网格面上的节点值进行线性插值得到的。对于等值点，其节点值则是用包含此点的网格单元的所有节点进行线性插值得到的。

16.2 速度报告选项

FLUENT中速度矢量有下列几种表现形式：

（1）直角坐标速度：直角坐标系中的速度是最常用的一种速度形式。这些速度矢量用直角坐标系三个坐标方向的分量加以定义。选择这种速度表达形式时要选择X Velocity（X方向速度分量），Y Velocity（Y方向速度分量）或Z Velocity（Z方向速度分量）。

（2）柱坐标系速度：这些速度矢量是由轴向速度分量、径向速度分量和切向速度分量唯一确定的，且基于以下坐标系而建立：

1）在轴对称问题中，旋转轴必须是x轴，x轴方向是轴向，y轴方向是径向。如果所计算的流场是轴对称旋转流场，那么旋转方向就是切向。

2）在仅包含一个网格区域的二维问题中，垂直屏幕的方向（z轴方向）是轴向，坐标原点由用户在Fluid面板中设定。

3）在仅包含一个网格区域的三维问题中，在Fluid面板中指定旋转轴和坐标原点，就可以完成坐标系的定义。

4）在多重区域计算中，例如多重参考系或滑动网格问题中，只要在Fluid或Solid面板中指定参考域的旋转轴，就可以完成坐标系的定义。参考域在Reference Values（参考值）面板中选择。

在上述圆柱坐标系的定义中，径向速度的正向以旋转轴为中心呈放射状指向外面，轴向速度的正方向和旋转轴矢量的方向一致，切向速度的正向则可以用右手法则来进行判断。

在查看计算结果时，可以选择Axial Velocity（轴向速度）、Radial Velocity（径向速度）等选项，以显示圆柱坐标速度。图16-2给出了不同类型域中的圆柱坐标速度：对于三维问题，可以报告轴向速度、径向速度和切向速度。对于二维问题，可以报告径向速度和切向速度。对于轴对称问题，可以报告轴向速度和径向速度。如果计算问题中采用了轴对称旋转模型，那么也可以给出旋转速度（它等于切向速度）。

（3）相对速度：这些速度的大小和方向取决于坐标系的选择和移动参考系的运动。在使用旋转参考系、混合面模型、多重参考系或滑动网格模拟流体时，相对速度很有用。要在结果中给出相对速度，需要选取Relative X Velocity（相对X向速度）、Relative Y Velocity（相对Y向速度）和Relative Z Velocity（相对Z向速度）等。相对速度既可以采用直角坐标形式，也可以采用圆柱坐标形式。

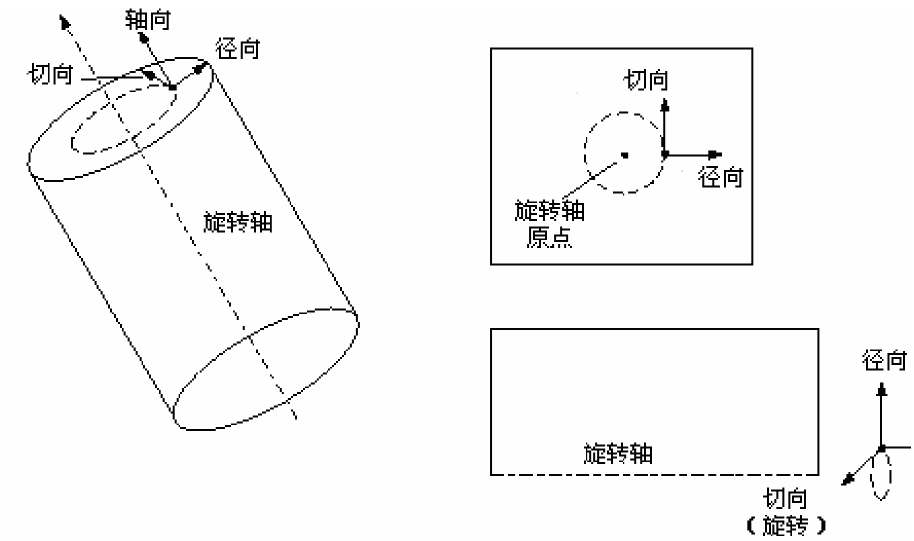


图 16-2 在三维、二维和轴对称域中的圆柱坐标速度分量

如果使用的是单一的旋转参考系，则相对速度就是针对这个移动参考系的相对速度。如果使用的是多重参考系、混合面模型或滑动网格，那么需要在Reference Values（参考值）面板中指定一个参考系（Reference Zone）用于计算相对速度。每个网格区域的旋转轴都可以在与之相对应的Fluid（流体）面板或Solid（固体）面板中定义。

当然，如果计算中没有涉及到任何运动的区域，则相对速度和绝对速度是相等的。

相对速度还可以用来计算驻点参数（总压和总温），上面谈到的圆柱坐标系也可以用来定义轴向坐标（Axial Coordinate）和径向坐标（Radial Coordinate）。

16.3 定制场函数

FLUENT除了提供基本的场变量外[1]，用户还可以定义自已的场函数，并与使用这些变量的命令（例如，display contour、display vector等）配合起来应用。这种功能可以在Custom Field Function Calculator（定制场函数）面板中实现。用户可以使用默认的场变量、已经定义的计算器函数和计算器算子创建新函数。在保存算例文件时，所定义的任何场函数都将保存在这个算例文件中，但是也可以把用户定义场函数保存在一个单独的文件中，以便在调用不同的算例文件时使用它。

请注意，所有的用户定义场函数都是在SI单位中求解和存储的。如果在场函数的定义中所用求解器定义的流场变量不在SI单位中，那么它们将被自动地转换，但是用户必须在适当的单位中认真地输入常数。还需要注意的是，显式节点值在自定义函数中不可用。如前所述，这些函数的节点值是通过对周围所有单元值进行平均得到的。

16.3.1 创建自定义场函数

执行下列菜单操作可以启动 Custom Field Function Calculator（定制场函数计算器）面板，如图 16-3 所示：

Define -> Custom Field Functions...

定义场函数需要用到场变量和计算器按钮，其操作过程与使用普通电子计算器非常类似。

[1] FLUENT 的场变量清单可以在 FLUENT 公司提供的用户手册中找到，这里不再一一列举。

其使用步骤大致如下：

（1）在场函数列表中选择一个场变量。在 Field Functions（场函数）下面有两个下拉列表，上面一个是函数类列表，下面一个是函数列表。首先在函数类中确定场函数所属的类，比如欲选择静压则先选择其类 Pressure...（压强），此时下面的函数列表将自动出现压强类的函数，其中就包括静压。选择 Static Pressure（静压），然后再点击 Select（选择）按钮，则静压的函数名称 p 将出现在 Definition（定义）下面的文字栏中。

（2）选择操作符，即在左面的按钮中选择代表计算符的按钮，点击按钮则按钮名称将出现在 Definition（定义）下面的文字栏中。重复（1）、（2）二个步骤，就可以得到新函数的定义式。这些操作符按钮都与计算器上的标志相同，需要特别提醒的是：DEL 按钮是删除按钮，点击这个按钮，光标将回退一格，清除最后一次的输入；CE/C 按钮是清除按钮，点击这个按钮将彻底清除函数定义和函数名。

如果想获得反三角正弦函数、反三角余弦函数或反三角正切函数，那么需要点击INV按钮，然后再选择sin，cos或tan。

点击ABS按钮后输入一个数，那么产生的将是这个数的绝对值，点击log按钮后产生的将是随后输入的这个数的自然对数。

PI按钮表示圆周率π，e按钮表示自然对数系统的底（它约等于2.71828）。

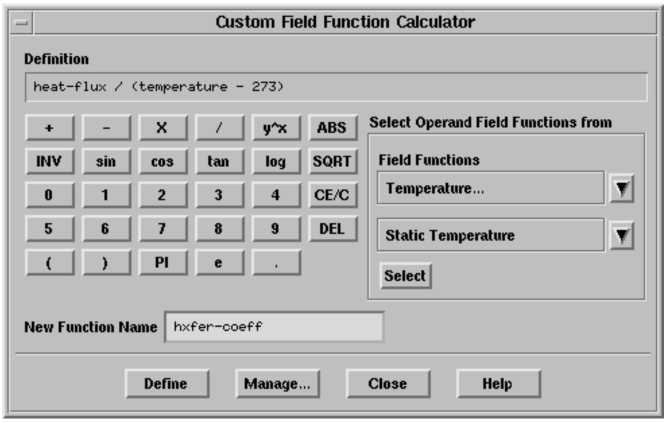


图 16-3 Custom Field Function Calculator（定制场函数计算器）面板

（3）最后在 New Function Name（新函数名）中为新函数设定一个名称，并点击 Define（定义）按钮，则完成场函数的定义。

16.3.2　操作、保存、加载自定义函数

一旦定义了自已的场函数，就可以在Field Function Definitions（场函数定义）面板中对它们进行操作。在Custom Field Function Calculator（定制场函数计算器）面板中点击Manage（管理）按钮，就可以打开这个面板，如图16-4所示。

在Field Function（场函数）清单中选择一个场函数名，可以展示这个函数的定义，以便确定它是否正确。如果确认这个函数定义不正确，并且需要重新定义或给函数一个新名字，那么可以点击Delete（删除）按钮删除这个函数，然后返回Custom Field Function Calculator（定制场函数计算器）面板重新进行定义。还可以把自定义场函数存放在一个文件里，或者从一个文件中读取它们。自定义场函数允许用户在算例文件之间传递自定义函数。

若要重命名一个函数，则在Field Functions（场函数）清单里选择它，然后在Name（名称）下面的文字框里输入一个新名字，最后点击Rename（重新命名）按钮。确定这个函数名没有和已经应用着的标准场函数名一样（例如，velocity-magnitude）。通过选择display/contour文本命令，

并查看其中可用选项，可以看到FLUENT中一列完整的预先定义好的场函数。

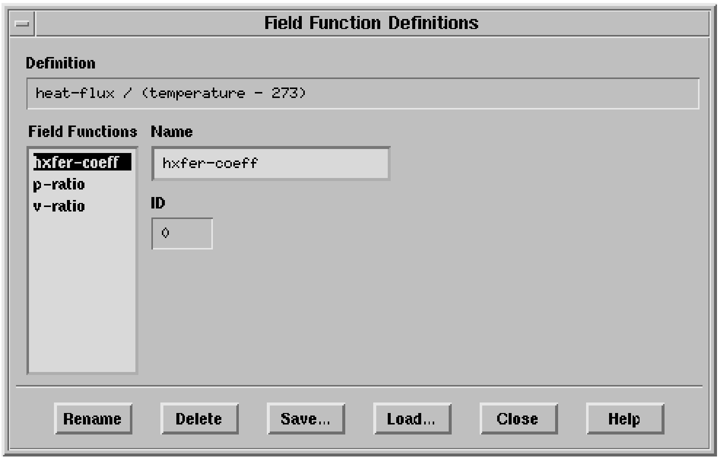


图 16-4 Field Function Definition（场函数定义）窗口

若要把Field Functions（场函数）列中的所有函数保存到一个文件中，则点击Save…（保存）按钮，并且在结果Select File（文件选择）对话框里给出这个文件的名字。

若要从一个保存过的文件中读取自定义场函数，则点击Load…（加载）按钮，并且在结果Select File（文件选择）对话框中指出这个文件的名字。自定义场函数是有效的Scheme函数，也可使用File/Read/Scheme…菜单项装载。

16.3.3　自定义场函数实例

当检查数值模拟结果的时候，用户会发现定义类似于下面的一些自定义场函数是非常有用的。

1.定义一个确定静压与进口总压比率的函数，应用关系

$$R = \frac{p + p_{op}}{p_{to} + p_{op}} \qquad (16\text{-}1)$$

其中，p 是求解器计算所得的静压，p_{to} 是进口总压，p_{op} 是问题的工作压力。p 是使用求解器定义的静压（Static Pressure），p_{to} 的数值是在Pressure Inlet面板中的Gauge Total Pressure规定的。规定工作压力 p_{op} 的值等于在Operating Conditions面板中所设定的值。如果工作压力是零，作为可压缩流动计算的一般情况，压力比率表达式退化为：

$$PR = \frac{p}{p_{to}} \qquad (16\text{-}2)$$

2.定义一个决定临界速度比率 v/a_o 的函数，它是一个参数，有时在涡轮机械计算中有应用，它的表达式为：

$$\frac{v}{a_o} = \left[\left(\frac{\gamma+1}{\gamma-1}\right)\left(1 - PR^{(\gamma-1)/\gamma}\right)\right]^{1/2} \qquad (16\text{-}3)$$

在这个关系式中，a_o 是临界速度，γ 是比热率，PR 是（16.5.2）式中定义的压力比率。

对于比热率γ，在特性…类中选择比热率（γ）。为包括PR，在Field Functions下的第一个下拉清单中选择Custom Field Functions…，然后在第二个清单中选择函数的名字PR（你定义过的）。

3.假设在一个管道里沿z轴有漩涡流动，要计算通过一横截面的角动量流动率：

$$\int \rho r v_\theta \vec{v} \cdot d\vec{A} \quad (16\text{-}4)$$

可以建立一个关于rv_θ的函数，这里r是径坐标，v_θ是切速度。然后应用Surface Integrals（表面积分）面板来计算角动量流动率。

第 17 章　并行处理

FLUENT 有串行求解器和并行求解器，串行求解器不能胜任一些大型工程项目的计算，为解决此问题，FLUENT 提供了并行求解器。本章将在以下 4 节中介绍并行处理的特性。

17.1：并行处理简介

17.2：启动求解器的并行版

17.3：使用工作站的并行网络

17.4：检查并改进并行计算性能

17.1　并行处理简介

FLUENT 串行求解器在一个 CPU 上用一个进程处理计算任务，并行求解器则是在多个 CPU 上用多个进程处理计算任务。这多个进程既可以在同一台计算机的多个 CPU 上运行，也可以在用网络连接的不同计算机上运行。图 17-1 和图 17-2 分别是 FLUENT 的串行求解器和并行求解器各模块之间的逻辑关系。

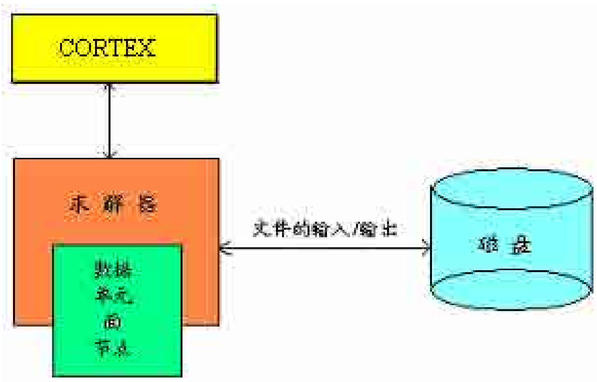


图 17-1 串行求解器各模块之间的逻辑关系

FLUENT 中的并行处理过程包括 FLUENT、一个主机进程和一套节点进程之间的交互作用。FLUENT 通过一个叫 CORTEX 的模块与主机进程和各个节点进程进行信息传递。CORTEX 主要用来管理 FLUENT 的用户界面和基本的图形功能。

FLUENT 并行求解器把网格和数据分成多个分区，然后再把各个分区分配给计算节点进行计算。分区的数量应该等于可用计算节点数量的整数倍。例如，如果将网格和数据划分为 8 个分区，则可以在计算中使用 1、2、4 或者 8 个计算结点。节点进程可以在大型并行计算机上运行，也可在一个多 CPU 的工作站上运行，或者在由多个计算机组成的网络上运行。

大体上说，计算节点数目越多，并行计算的时间就越短。然而节点数量与计算效率之间并不是存在严格的正比关系。随着节点间数据通信量的增加，并行效率将会降低，所以在选择使用并行计算之前，应该思考是否确实需要使用并行计算方式。

FLUENT 的主机进程中没有任何网格数据，而仅仅负责解释来自于 CORTEX 的命令。主机进程把接收到的命令通过指定计算节点上的数据通讯模块传递给其他节点，这个被指定的节点称为“计算 0 节点”。其它计算节点在接收到主机进程传递过来的命令后，在各自的数据集合上同时执行相同的程序。计算节点与主机进程之间的通讯工作仅当计算 0 节点和所有计算节点保

持同步时才可以进行。

事实上每个计算节点和其它每一个计算节点是连接在一起的，但是节点之间的数据传输、同步和执行全局任务则需要通过其数据通讯模块实现，并通过数据通讯模块实现逻辑意义上的联通。FLUENT 的数据通讯模块是一个信息传送库，而信息的传送机制则符合 MPI（Message Passing Interface，信息传送界面）标准，如图 17-2 所示。

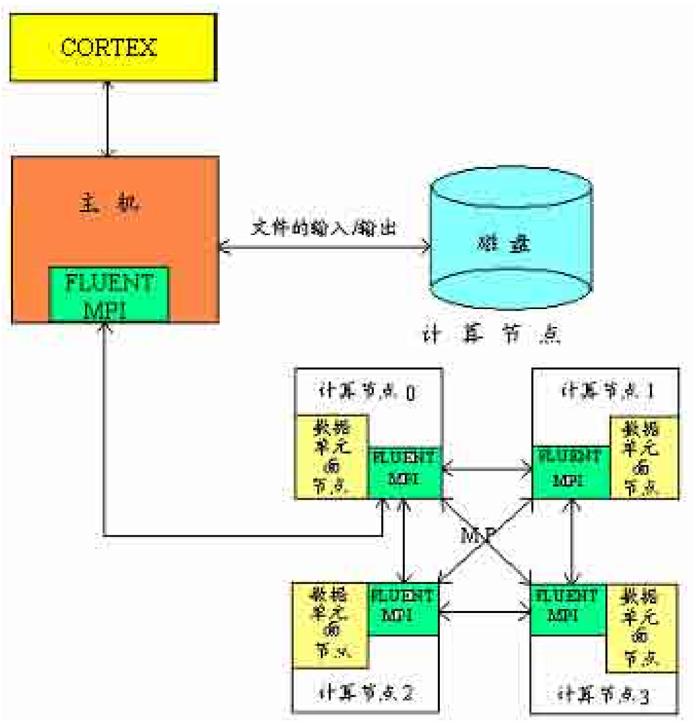


图 17-2 并行求解器各模块之间的逻辑关系

FLUENT 的并行处理进程和串行处理进程一样，均使用唯一的一个整数标志符加以识别。主机从计算 0 节点收集信息，然后执行关于所有数据的任务，包括打印、信息显示和文件写入等等，其方法与使用串行求解器时一样。

FLUENT 并行求解器的使用流程如下：

（1）启动并行求解器，并在需要的时候生成附加的计算节点。

（2）读取算例文件，并让 FLUENT 在加载文件时对网格进行自动分区。因为分区过程在某些情况下取决于计算模型的设定，因此网格分区最好在模型设置完毕后再去执行。除了自动分区方法，还有其它方法可以用来划分网格分区，其中包括在串行或并行求解器中的手动划分方式。

（3）仔细检查网格分区，如果分区存在问题，可以重新进行分区操作。

（4）开始进行计算。

17.2 启动求解器的并行版本

在采用多处理器设计的并行机和工作站集群上启动 FLUENT 并行版本的方法是不同的。FLUENT 并行求解器运行的环境既可以是 UNIX 操作系统，也可以是 LINUX 操作系统，还可以是 Windows 系统。在这里只详细介绍它在 Windows 操作系统下的情况。

17.2.1 在多处理器 Windows 操作系统的机器上运行

在 Windows 系统中，可以从 MS-DOS 命令提示窗口启动 FLUENT 的专用并行版本，启动命令为：

fluent *version* –tx

在实际使用过程中 *version* 用 FLUENT 的版本号（2d、2ddp、3d 或 3ddp）代替，x 为计算中使用的处理器数量。例如，命令 fluent 3d –t3 表示在 3 个处理器上运行 3D 版本的 FLUENT。

17.2.2 在一个 Windows 集群上运行

在由 Windows 机组成的网络上运行并行 FLUENT 可以使用两种通讯方式，一种是基于 RSHD 的并行版，另一种是基于 MPI 的并行版。下面分别介绍其启动方式：

1、起动基于 RSHD 的 FLUENT 并行版本。

如果网络中安装用于网络通信的 RSHD 软件，则在 MS-DOS 命令提示窗口中键入命令：

fluent *version –tnprocs* –pnet [-cnf=*hostfile*] –path*sharename*

其中：

（1）version 由 FLUENT 版本号（2d、2ddp、3d 或 3ddp）来代替。

（2）–pathsharename 为 FLUENT 的安装目录 Fluent.Inc 在网络上的共享名称。例如，如果在 Computer1 上安装了 FLUENT，那么就用\\Computer1\Fluent.Inc 替换 *sharename*。

（3）–cnf=*hostfile*（可选的）指定主机文件，主机文件中是运行并行计算的计算机名单。如果主机文件不在当前目录中，则需要提供文件的完整路径。在使用-cnf 选项时，不要使用中括号。

可以应用 Notpad 创建主机文件。对文件名唯一的限制是其中不能有空格符，比如可以使用 host.txt 做主机文件名，但不能用 my host.txt 做主机文件名。主机文件中可以包含如下的条目：

computer1
computer2

其中排在第一位的计算机是本地计算机名，最后一个条目的后面必须是一个空行。

如果网络中的有一台多处理器计算机，则主机文件中可以多次列出它。例如，如果 computer1 有两个 CPU，那么为了利用这两个 CPU，hosts.txt 文件应当列出 computer1 两次：

computer1
computer1
computer2

如果不选-cnf 项，FLUENT 将在键入启动命令的本地计算机上起动 nprocs 程序。然后，再用 FLUENT 中的 Network Configuration（网络设置）面板在集群上交互式地将其他计算机加入，具体操作见 17.3 节。

（4）–tnprocs 用于指定计算进程的数量。如果使用了-cnf 选项，那么 *hostfile* 中的计算机列表就用于指定哪些计算机将参与并行计算。例如，在主机文件里列出了 10 台计算机，如果仅想在 5 个计算机上进行计算，则设置 nprocs 为 5，即-t5，FLUENT 将用主机文件中的前 5 台机器进行计算。

在启动后可以用 Network Configuration（网络设置）面板删除进程或者添加进程，详细内容请看第 17.3 节。

下面是一个例子：主机文件 hosts.txt 里列出了 m 台（$m \geq 3$）计算机，在前 3 台上起动一个基于 RSHD 的 3D 并行工作，其完整命令如下：

fluent 3d –t3 –pnet –cnf=hosts.txt –path\\computer1\fluent.inc

2、起动基于 MPI 的 FLUENT 并行版本。

如果用于网络通讯的软件为 MPI 软件，则在 MS-DOS 命令窗口中应该键入如下命令：
fluent *version* –t*nprocs* –p*comm* -cnf=*hostfile* –path*sharename*
其中，*comm* 可以是 nmpi 也可以是 vmpi。这里 nmpi 意为 network MPI，即网络上免费发行的 MPI 软件；vmpi 意为 vendor-supplied MPI，即开发商提供的专用 MPI 软件。其余选项的含义与前面基于 RSHD 的启动里所讲的相同，所不同的是：

（1）必须使用主机文件。当应用 MPI 软件时，用 Network Configuration（网络设置）面板既不可以添加也不可以关掉集群上的计算节点。

（2）主机文件中列出的第一台计算机必须是本地计算机的名字。

作为一个例子，主机文件 hosts.txt 里列出了 m 台（$m \geq 3$）计算机，在前 3 台上起动一个基于 MPI 的 3D 并行工作的完整命令如下：
fluent 3d –t3 –pvmpi –cnf=hosts.txt –path\\computer1\fluent.inc

17.3　使用并行的工作站网络

在工作站网络上通过添加和删除计算节点可以创建一台虚拟的并行机，即可以将网络看作一台拥有多个 CPU 的并行计算机，然后在这台虚拟机上进行并行计算。不过多个 CPU 并非进行并行计算的必要条件，事实上即使在一台只有一个 CPU 的工作站上也可以运行多个计算节点进程，以并行方式进行计算。

17.3.1 设置网络

添加和删除计算节点的工作可以在 Network Configuration（网络设置）面板中进行，启动这个面板的菜单操作为：
Parallel -> Network -> Configure...
启动后面板如图 17-3 所示。

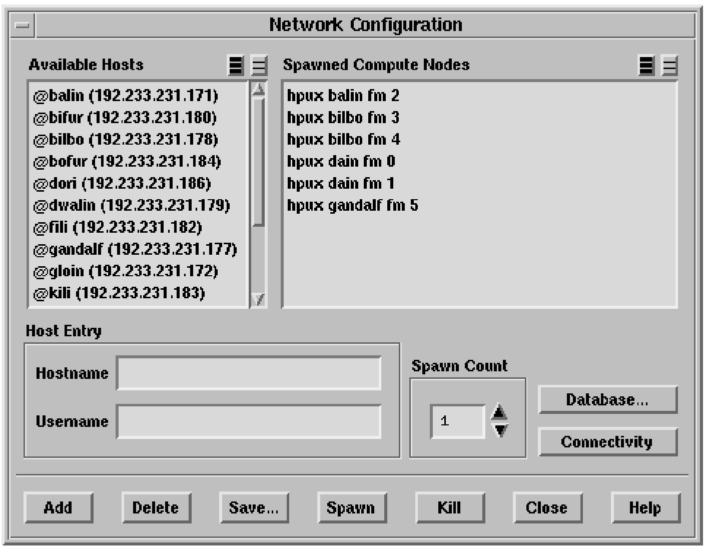


图 17-3 Network Configuration 面板

1、网络的结构

计算节点的排序从 0 开始。除了计算节点进程，还存在一个主机进程。这个主机进程在 FLUENT 启动时自动启动，在 FLUENT 退出时则被自动关掉。除主机进程外，其他节点进程都可以在计算进行时被关闭，其中第 0 个节点上的节点进程仅在其他节点进程被全部关闭后，才能被关闭。主机进程不能在计算过程中被关闭。主机进程在本地计算机上创建计算 0 节点，计算 0 节点再依次创建其它计算节点。

2、创建计算节点的步骤

创建计算节点的基本步骤如下：

（1）在 Available Hosts（可用主机）列表中选择要创建计算节点的计算机。如果想要的计算机不在列表中，可以使用 Host Entry（主机入口）下面的空格手工添加一台主机，或者从主机数据库（Datebase…）里将想要的主机拷贝进来。

（2）在 Spawn Count（创建计数器）空框里设置每一台主机上需要创建的节点进程的数量。

（3）点击 Spawn（繁衍）按钮完成新节的创建过程，同时节点名称被添加到 Spawned Compute Nodes（创建计算节点）列表中。

下面将介绍与网络结构相关的其余功能。

3、手工添加主机

要在 Network Configuration（网络设置）面板中手工添加一个主机到 Available Hosts（可用主机）列表，需要在 Host Entry（主机入口）下面的 Hostname（主机名称）里输入远程机器的网络名称，然后在 Username（用户名）里输入用户名称。如果所有帐号的用户名都相同，则不需要输入用户名。最后点击 Add（添加）按钮，则主机将被添加到 Available Hosts（可用主机）列表中。

4、删除主机

要在 Network Configuration（网络设置）面板中的 Available Hosts（可用主机）列表里删除一个主机，那么选取这个主机后点击 Delete（删除）按钮即可完成。可以看到主机的名字将从 Available Hosts（可用主机）列表中消失，不过主机数据库不会因为这项操作发生任何变化。

5、关闭计算节点

如果需要关闭一个计算节点，可以在 Spawned Compute Nodes（已创建节点）列表中选择它，然后点击 kill（关闭）按钮就可以完成关闭节点的操作。需要再次提醒的是，计算 0 节点仅当是最后剩余的一个计算节点程序时才可以被关闭。

6、保存主机文件

如果已经完成 Available Hosts（可用主机）的创建工作，可以将含有主机信息的主机文件进行保存，以便再次使用同一组主机进行计算时可以不必再重新进行创建工作。点击 Save（保存）按扭，然后在 Select File（选择文件）对话框里输入相应的文件名并保存它就可以了。在将来需要用到这个文件时，可以把这个文件的内容载入主机数据库，然后再将这些主机复制到 Network Configuration（网络设置）面板，就可以快速生成 Available Hosts（可用主机）列表了。

7、节点创建过程中遇到的一般问题

创建过程中的节点进程将试图与一个新的计算节点建立联系，但是如果 50 秒后没有收到这个新节点的响应，那么进程将认为创建过程失败。例如，如果远程访问的计算机不能找到可执行的 FLUENT，那么创建进程的操作既告失败。如果想用人工方法验证准备执行节点创建的计算机能否启动一个新计算节点，可以键入下列命令：

rsh [-1 *username*] *hostname* fluent –t0 –v

hostname 应当用计算机的网络名称代替，*usename* 应当被指定为 *hostname* 的远端机器的注册名所取代。

!如果你所有的记述都有相同的注册名，那么你不需要规定用户名。（中括号里面的-1 *username* 意味着不是总需要；如果你没有输入一个注册名称，那么就不存在中括号及里面的内容。）注意：在一些系统上，远端设定命令的解释器位置的命令是 remsh 而不是 rsh。

繁衍验证可能失败的几个原因为：

注册错误。机器繁衍一个新计算节点必须能够 rsh 到这台将拥有新程序的机器，否则繁衍将会失败。使其具备这种功能的方法有好几种，请访问你的系统管理器寻求帮助。

fluent：找不到命令。rsh 到远端机器取得了成功，但在这台机器上找不到 FLUENT 设定命令的解释器位置的 script 路径。如果你使用的是 csh，那么 FLUENT 设定命令的解释器位置的 script 路径应当添加路径变量到你的.cshrc 文件中。如果这也失败了，你可以在创建计算节点之前，直接使用 parallel/network/path 文本命令设置路径到 Fluent.Inc 安装地址里。

17.3.2 主机数据库

当你要创建一个工作站的并行网络时，在你当地网络（一个 “主机文件”）中的部分机器基础上来进行这个工作是一个比较便捷的方法。你可以卸载一个含有这些机器名字的文件到主机数据库，然后利用 **Hosts Database** 面板，选择用于创建一个工作站簇上的并行结构（或网络）的主机（图 17.3.2）。

Parallel → Net work → … Database



图 17.3.2：Hosts Database 面板

你也可以在 **Network Configuration** 面板里点击 **Database…**按扭打开这个面板。

如果主机文件 fluent.hosts 或.fluent.hosts 在你的主目录里，它的内容是在启动的时候自动添加到主机数据库中的。另一方面，主机数据库将是空的，直到你读进一个主机文件为止。

读主机文件

如果你有一个含有你当地网络上的一列机器的主机文件，你可以通过点击 **Load…**按扭卸载这个文件到 **Hosts Database** 面板里，在 **Select File** 对话框里规定这个文件的名字。一旦读了这个文件的内容，这个主机名字将会出现在 **Hosts** 列中。（FLUENT 将自动地为每一台识别出的机器添加 IP（网络协议）地址。如果一台机器目前不在当地网络中，那么它的标示将是 **unknown**。

复制主机到 **Network Configuration** 面板

如果你要在 **Hosts Database** 面板中复制一个或多个主机到 Network Configuration 面板中的 **Available Hosts** 列，那么在 **Hosts** 列中选择你想选的名字，然后点击 **Copy** 按扭。所选主机将被添加到 **Available Hosts** 列，在这些主机上你可以繁衍节点。

检查网络的连通性

对于任何计算节点，你都可以打印网络连通性的信息，其中包括主机名字、体系机构、程序标识符和所选计算节点的标识符以及所有联网的机器。所选计算节点的标识符之记号是星号。

FLUENT 主机程序的标识符一直是 host。计算节点从 node-0 开始按数字的大小顺序依

次标识。所有的计算节点完全连接了起来。另外，计算 0 节点连接主机程序。

为获得一个计算节点的连通性信息,你可以使用 **Parallel Connectivity** 面板(图 17.3.3)。

Parallel → Show Connectivity…

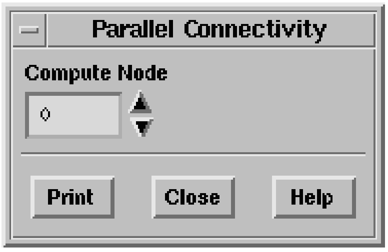


图 17.3.3：Parallel Connectivity 面板

为获得 **Compute Node** 下面框里的连通性信息，那么显示计算节点标识符，然后点击 **Print** 按扭。计算 0 节点作为样本输出如下：

标识符	Comm.	主机名	O.S	PID	Mach ID	HW ID	名字
host	net	balin	Linux-32	17272	0	7	Fluent Host
n3	smpi	balin	Linux-32	17307	1	10	Fluent Node
n2	smpi	filio	Linux-32	17306	0	-1	Fluent Node
n1	smpi	bofur	Linux-32	17305	0	1	Fluent Node
on	smpi	balin	Linux-32	17273	2	11	Fluent Node

O.S 表示体系机构，Comm.表示交流器，PID 表示程序标识数字，Mach ID 表示计算节点的标识符，HW ID 表示所用交流器的特定标识符。

你可以在 **Network Configuration** 面板里检查一个计算节点的连通性，具体过程是：在 **Spawned Compute Nodes** 列选择这个计算节点，然后点击 **Connectivity** 按钮。如果在 **Spawned Compute Nodes** 列中你没有做任何选择就点击了 **Connectivity** 按钮，那么 **Parallel Connectivity** 面板将会被打开,你可以如上面所叙述的方法在此规定节点。如果你在 **Spawned Compute Nodes** 列中选择了不止一个计算节点，那么点击 **Connectivity** 按钮将会打印出每一个所选节点的连通性信息。

16.4　检查并改进并行计算性能

为了确定并行求解器工作运行的好坏,你可以使用性能仪表来测量计算和通信的时间,总并行效率。为了优化并行求解器，你可以控制节点间的通信总量和提高 FLUENT 的自动负载平衡特性。

17.4.1 检查并行计算性能

性能仪表不仅允许你报告信息传送的统计数据，还允许你报告一个计算过程中的严密同步时间。自从性能仪表一直工作起，你可以在计算完毕后通过输出这些统计资料而访问他们。若要浏览当前的统计信息，则使用 **Parallel/Timer/Usage** 菜单项。

Parallel → Timer → Usage…

性能统计数据将在文本窗口里输出（控制台）。

为了刷新性能仪表以便你可以从以后的报告中清除以前的统计数据，则使用 **Parallel/Timer/Reset** 菜单项。

Parallel → Timer → Reset…

17.4.2 优化并行求解器

增加报告的时间间隔

在 FLUENT 中，你可以对残差的输出/绘图或其它解答的监控增加报告时间间隔，从而降低通信和提高并行计算的性能。你可以在 **Iterate** 面板中修改 **Reporting Interval** 的值。

Solve → Iterate…

!注意：直到每一报告间隔的最后你才能够打断迭代。

负载平衡

在 FLUENT 中可以使用动力负载平衡性能。应用并行处理的主要原因是为减少数值模拟的周转时间，理想上是通过与计算所用机器的组合速度成比例的一个因数来实现。例如，如果你应用 4 个 CPU 去求解问题，那么你希望通过 4 的一个因数来减少计算的周转时间。当然，这是一个理想状态，要假设 CPU 之间所需要的交流通信很少，所有的 CPU 的速度有相同的速度，所有的 CPU 致力于你的工作中。实际上，情况并非如此。例如，若你在一个非同类机器集合的工作站上正在进行并行求解，CPU 的速度可以变化，其它工作可能使用一个或多个 CPU 就可完成，来自并行求解器内部或产生于外部资源的网络传输可能延迟 CPU 之间必要的通信。

如果你使 FLUENT 产生动力负载平衡，那么通过计算和联网的机器之载荷将被周期性的监控。如果负载权衡器认为通过计算节点间的单元重新分布可以改进性能，那么它将自动如此做。有一个与负载平衡本身相联系的时间损失，所以它不能够缺省。如果你将使用一台专用同类机器，或正在使用一台异类机器，但已考虑到了在规定一个负载分布的划分期间的 CPU 速度的不同，那么你可以不需要使用负载平衡。

！注意：当应用 shell 传导模型时，你将不能打开负载平衡。

若要赋予和控制 FLUENT 的自动负载平衡特性，使用 **Load Balance** 面板（图 16.4.1）。负载平衡将自动探测和分析并行性能，并在现存的计算节点之间重新分布单元从而优化它。

Parallel → Load Balance…

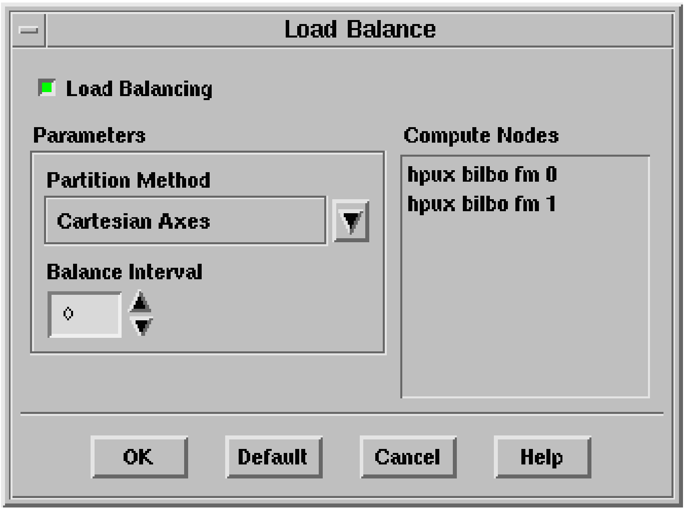


图 17.5.1；Load Balance 面板

使用负载平衡的步骤如下：

1. 打开 Load Balancing 项。
2. 在 Partition Method 下拉菜单中选择对分方法来建立新的网格分割。作为自动负载平衡过程的一部分，应用特定方法把网格重新划分成几个小的分割。然后所得到的分割将被分配到计算节点之中，从而获得一个更大的平衡负载。
3. 规定想要的 **Balance Interval**。当规定了一个 0 值时，FLUENT 将在内部决定使用的最佳值，初始使用的是一个 25 步迭代的间隔。你也可以不考虑这种行为而规定一个非零值。然后，FLUENT 将在没 N 步迭代后试着执行负载平衡，这里 N 是所规定的 **Balance Interval**。你在选择一个迭代时应当非常小心，因为这个迭代要大到超过执行负载平衡操作所花费的时间。

注意：你可在任何时候打断这个计算，关闭（或打开）负载平衡特性，然后继续此计算。

！如果由于适应而在你的计算中出现问题，你可以关掉自动负载平衡，这种情况发生在并行计算中执行网格适应的任何时间。为了指示求解器跳过负载平衡步，发布如下命令：

（disable-load-balance-after-adaption）

应用如下命令回到默认的行为：

（enable-load-balance-after-adaption）

第18章　用户自定义函数

用户自定义函数（User-Defined Function，UDF）亦称自定义函数，它允许对 FLUENT 进行个性化设置，这对充分发掘 FLUENT 在功能上的潜力意义深远。本章比较详细地介绍了在 FLUENT 中怎样写、编译和使用 UDF，并且还给出了有关 UDF 的例子。

主要包括以下几个方面的内容：

18.1 概论

18.2 写用户自定义函数（UDF）

18.3 解释、编译及连接用户自定义函数（UDF）

18.4 用户自定义函数（UDF）举例

18.1　概论

18.1.1　什么是用户自定义函数（UDF）

用户自定义函数（UDF）是用户使用 C 语言编写的函数，它可以被 FLUENT 求解器动态地加载，从而增强 FLUENT 软件的现有功能。UDF 是使用 Fluent.Inc 提供的 DEFINE 宏定义的。所定义的函数使用由 Fluent.Inc 提供的预定义宏和函数从 FLUENT 求解器中访问数据。每一个 UDF 都要在程序的开头包含 udf.h 文件（#include ”udf.h”），使得在程序编译过程中，DEFINE 宏以及其它由 FLUENT 提供的宏和函数所定义的内容都能被引用。在 FLUENT 中，UDF 作为解释的或编译的函数被执行。不论是 UDF 传送到求解器的值，还是被求解器返回到 UDF 的值，都必须使用国际单位制。

UDF 的特征可以概括如下：

（1）是用 C 语言写成的。

（2）必须有一个关于 udf.h 文件的包含声明。

（3）必须使用由 Fluent.Inc 提供的 DEFINE 宏定义。

（4）使用由 Fluent.Inc 提供的预定义宏和函数访问 FLUENT 求解器的数据。

（5）作为解释的或编译的函数被执行。

（6）计算中必须使用国际单位制。

在 FLUENT 中，UDF 可以执行各种不同任务。如果它们在 udf.h 文件中没有被定义为 void，那么它们可以返回一个值。如果它们没有返回一个值，它们还可以修改一个哑元，或修改一个没有被作为哑元传递的变量，或者借助算例文件和数据文件执行输入输出任务。

概括说，UDF 能够完成下列任务：

（1）返回一个值。

（2）修改一个哑元。

（3）返回一个值同时修改一个哑元。

（4）修改一个没被作为一个哑元进行传递的变量。

（5）从算例文件或数据文件中写入或读出信息。

UDF 可以使用 C 语言在任何文本编辑器中进行编辑，但是源文件只能以扩展名.c 保存。通常源文件中只有一个 UDF 函数，但是也允许在一个文件中包含多个前后相连的 UDF 函数。源文件在 FLUENT 中既可以被解释也可以被编译。对于解释型的 UDF，源文件（例如，

pressure-profile.c）将直接被加载和解释。而对于编译型的 UDF，则首先要建立一个共享的目标模块库，然后再把它加载到 FLUENT 中。一旦解释或编译了 UDF，相应的 UDF 名字将会在 FLUENT 的图形界面中出现，并且在相应的面板中通过选择这个函数将其连接到一个求解器中。

18.1.2　使用 UDF 的原因

UDF 允许对 FLUENT 进行特殊设置以便满足建模的特殊需要。UDF 可以用于各种实际应用，其中包括：

（1）重新设置边界条件、材料特性、表面和体积反应速率、FLUENT 输运方程中的源项，自定义标量（UDS）输运方程中的源项和扩散函数等等。

（2）在每次迭代计算后调整计算结果。

（3）解的初始化。

（4）根据计算要求，非同步执行 UDF。

（5）加强后处理过程。

（6）加强 FLUENT 中现有的物理模型（例如，离散相模型、多相混合物模型、离散点辐射模型）。

18.1.3　UDF 的局限性

在 FLUENT 中，虽然 UDF 的性能涉及到非常广的应用范围，但 UDF 不可能适用于每一个应用。毕竟不是所有的解变量或物理模型都可以被 UDF 访问。比如，比热值不能被修改，若要修改，则需要附带的求解器功能。如果对一个特定问题不能确定是否可以使用 UDF 来处理，那么可以联系有关技术支持工程师寻求帮助。

18.1.4　网格术语

大部分 UDF 是从 FLUENT 求解器访问数据的。由于求解器数据是根据网格而定义的，因此在写一个 UDF 之前，需要掌握基本的网格术语。

一个流场的网格是由大量控制体或者单元构成的。每个单元是由一组网格点（或节点），一个单元中心和包围这个单元的面定义的（如图 18-1 所示）。FLUENT 使用内部数据结构定义网格的域，并将单元、单元面和网格中的节点按序排列，从而建立相邻单元之间的连通性。线程是 FLUENT 中数据结构的内部名称，它用来表示一个（边界或单元）区。单元线程是单元的组合，面线程是面的组合。域是 FLUENT 中数据结构的内部名称，它被用来表示网格中的节点、面和单元线程的组合。

单元和单元面组成代表流场各个部分的区域（进口、出口、壁面、流动区域等等）。一个面围住的是一个还是两个单元（用 c0 和 c1 加以识别）取决于它是一个边界面还是一个内部面。如果是边界面，那么仅有 c0 存在（对于一个边界面，不定义 c1）。如果是内部面，则需要同时定义 c0 和 c1 单元。一个面两侧的单元可以属于同一个单元线程，也可以属于另一个单元线程。

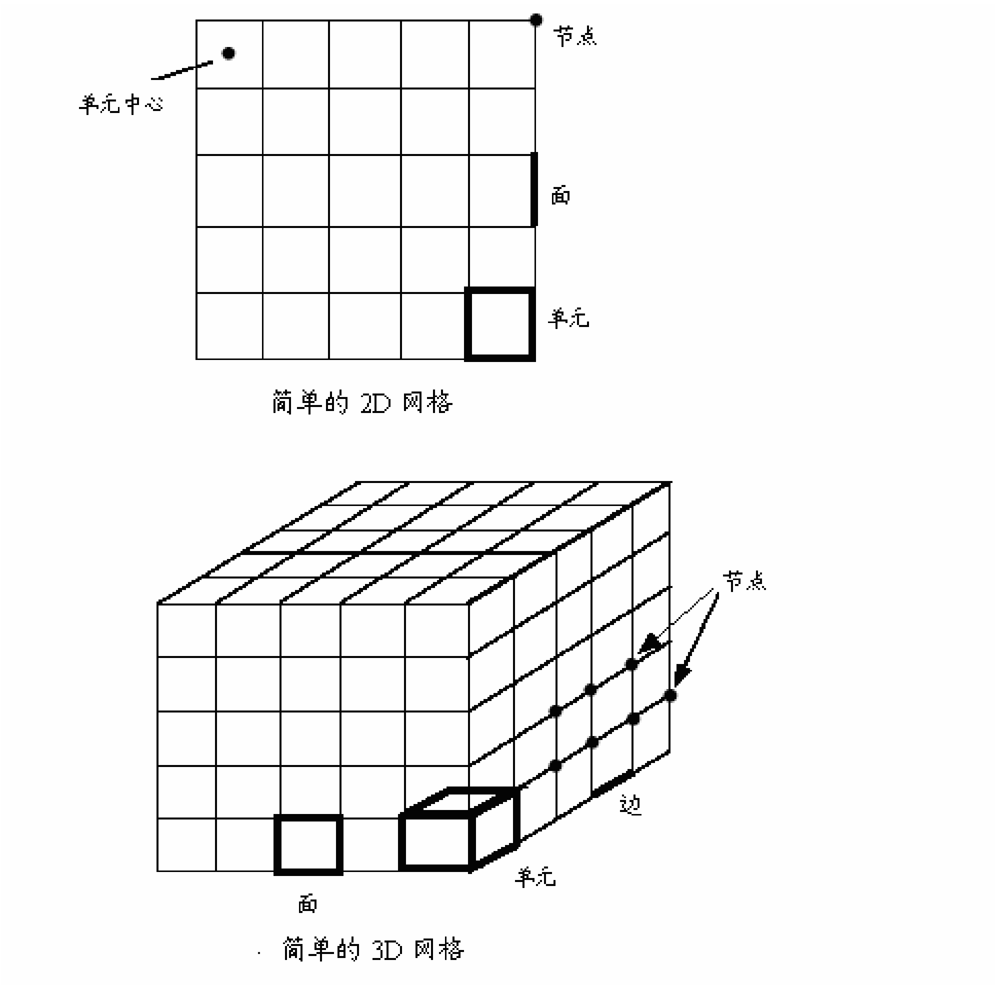


图 18－1：网格组分

单元	包含控制体的分离域
单元中心	存储单元数据的位置
面	一个单元的边界（2D 或 3D）
边	一个面的边界（3D）
节点	网格点
单元网络	单元的组合
面网络	面的组合
节点网络	节点的组合
域	节点，面和单元线程的组合

18.2　写用户定义函数（UDF）

UDF 是用 FLUENT 软件中提供的 DEFINE 宏加以定义的。本节将从以下几方面对 DEFINE 宏展开叙述：

18.2.1: 引言
18.2.2: 一般 DEFINE 宏
18.2.3: 物种模型的 DEFINE 宏
18.2.4: 多相的 DEFINE 宏
18.2.5: 动态网格的 DEFINE 宏
18.2.6: DPM DEFINE 宏

18.2.1　引言

使用 DEFINE 宏定义 UDF

UDF 是用 FLUENT 提供的函数声明语句来定义的，这些函数声明语句被包含在 DEFINE 宏中，而 DEFINE 宏的定义则包含在 udf.h 头文件里。DEFINE 宏的一般格式是：

DEFINE_MACRONAME(udf_name，passed-in variables)

其中，圆括号内的第一个哑元是 UDF 的名字，名字必须用小写字母定义。DEFINE 宏中的第二个哑元是从 FLUENT 求解器传送到 UDF 中的变量。例如宏：

DEFINE_PROFILE(inlet_x_velocity，thread，index)

定义了一个命名为 inlet_x_velocity 的型函数，另外的两个哑元，thread 和 index 是将从 FLUETN 传送进这个函数的变量。这两个被传进的变量分别是边界区域的标示符（即指向边界单元线程的指针）和用来标明被存储变量的指标。一旦编译了这个 UDF，函数名 inlet_x_velocity 将出现在 Velocity Inlet 面板的下拉菜单中供用户选择。

注意，一个 DEFINE 宏的所有自变量需要放置到源代码中的同一行。把这个 DEFINE 说明分成若干行将会导致编辑上的错误。

在源文件中加入 udf.h 头文件

udf.h 头文件不仅包含 DEFINE 宏的定义，还包含 C 语言库函数头文件的#include 编辑器指令，同时还包含其它由 FLUENT 提供的宏和函数的头文件（例如，mem.h），所以用户所写的每个 UDF 源代码的第一行必须是#include 编辑器指令和 udf.h 文件，即：

```
#include ”udf.h”
```

例如，在前面那个例子中，udf.h 被包含在其源文件中，

```
DEFINE_PROFILE(inlet_x_velocity，thread，index)
```

则在编译过程中，这个宏将扩展为：

```
void inlet_x_velocity(Thread*thread，int index)
```

在用户编辑 UDF 时，不需要把 udf.h 的副本放到用户的当地目录中。一旦编辑了一个 UDF，

FLUENT 求解器会自动地从系统目录中读入这个 udf.h 文件。

把所有的函数原型放到一个头文件中是一个常规做法。所以，如果 UDF 利用自定义函数，那么附加的自定义头文件可能也需要放到 udf.h 中。

所有 DEFINE 宏的定义都包含在 udf.h 头文件中。为了方便用户，把它们列在了附录 A 中。

DEFINE 宏的分类

DEFINE 宏基本上可分为五类：

（1）一般求解器

（2）物种模型

（3）多相

（4）动态网格模型

（5）弥散相模型（Discrete Phase Model，DPM）

对每一种 DEFINE 宏，在讲解过程中大部分都附有一个 UDF 的源代码例子。不过，请注意，不是所有的例子都是一个完整的函数，因为有些函数不能作为 FLUENT 中的单机函数执行。举例的意图仅仅是示范 DEFINE 宏的应用。

使用某个宏定义的 UDF，在被通译或编译之后，在 DEFINE 宏的自变量中规定的函数名将会在 FLUENT 其对应的面板中变地可视与可选，那么将需要把 UDF 连接到 FLUENT 中。第 18.3 节详细的介绍了通译和编译及连接用户定义函数。

18.2.2　一般 DEFINE 宏

一般 DEFINE 宏执行一般的求解函数，其中这些函数都是 FLUENT 中的独立模型。表 18.2.2.1 给出了一个关于 DEFINE 宏，DEFINE 宏定义的函数，以及 FLUENT 中被激活的面板的快捷参考指导。每一个 DEFINE 宏的定义都被放在了 udf.h 头文件中。为方便读者，把它们列在了附录 A 中。

一般 DEFINE 宏包括以下 5 种：

（1）DEFINE_ADJUST

（2）DEFINE_EXECUTE_AT_END

（3）DEFINE_INIT

（4）DEFINE_ON_DEMAND

（5）DEFINE_RW_FILE

表 18－1:关于一般求解器 DEFINE 宏的快捷参考指导

函数	DEFINE 宏	激活的面板
操作变量	DEFINE_ADJUST	User-Defined Function Hooks
在迭代或时间步的最后执行	DEFINE_EXECUTE_AT_END	User-Defined Function Hooks
初始化变量	DEFINE_INIT	User-Defined Function Hooks
非同步执行	DEFINE_ON_DEMAND	Execute On Demand
读/写变量到 case 和 date 文件	DEFINE_RW_FILE	User-Defined Function Hooks

1. DEFINE_ADJUST

DEFINE_ADJUST 是一个一般用途的宏，它用来调整或修改 FLUENT 的变量，且变量作为自变量而不被传送。例如，可以使用 DEFINE_ADJUST 修改流动变量（例如，速度，压力）和计算积分。也可以使用它对一个标量在一个域上进行积分，以及在积分结果的基础上对边界条件进行调整。一个使用 DEFINE_ADJUST 定义的函数在每步迭代都运行，并且这个函数是在输运方程被求解之前的每步迭代的开始时被调用。

- 应用

宏：	DEFINE_ADJUST (name, d)
自变量类型：	Domain *d
函数返回：	void

DEFINE_ADJUST 中有两个自变量：name 和 d。name 是由用户给出的 UDF 的名字。d 是被传送到 FLUENT 求解器的自变量，它是域的一个指示器，其中校准函数在这个域上被使用。域自变量提供到网格中所有单元的和面的线程之途径。对于多相流动，被求解器传送到这个函数的域指示器是平均混合物的域指示器。

- 例子

下面的这个名为 my_dajust 的 UDF，使用 DEFINE_ADJUST 对湍流耗散在整个域上进行积分。然后在控制台窗口中输出积分值。每步迭代 UDF 都会被调用。在 FLUENT 中，这个 UDF 可以作为一个通译的或汇编的 UDF 被执行。

```
/*********************************************************************
  UDF：对湍流耗散进行积分，在控制台窗口输出积分值
*********************************************************************/
   #include “udf.h”

   DEFINE_ADJUST(my_adjust, d)
   {
     Thread *t;
     /*对耗散进行积分. */
     real sum_diss=0.;
     cell_t c;

     thread_loop_c (t,d)
     {
         begin_c_loop (c,t)
             sum_diss += C_D(c,t)*
             C_VOLUME(C,T);
        end_c_loop (c,t)
     }
     printf(“Volume integral of turbulent dissipation:%g\n”,sum_diss);
   }
```

2. DEFINE_EXECUTE_AT_END

DEFINE_EXECUTE_AT_END 是一个一般用途的宏，它在一个稳态运行中的一个迭代的最后或者在一个暂时运行中的一个时间步的最后被执行。若想在这个特殊的时刻计算流量，则可以使用 DEFINE_EXECUTE_AT_END。注意，没有必要规定这个 UDF 是否在一个时间步的最后还是在一个迭代的最后获得执行。因为当用户在 FLUENT 中选择了稳定的或非稳定的时间方法时，这一工作是自动进行的。

- 应用

宏： DEFINE_EXECUTE_AT_END (name)
自变量类型： none
函数返回： void

DEFINE_EXECUTE_AT_END 中仅有一个自变量：name。name 是由用户给出的 UDF 的名字。与 DEFINE_ADJUST 不同的是，DEFINE_EXECUTE_AT_END 没有域指示器。所以，如果函数需要具有到一个域指示器的途径，那么将需要使用工具 Get_Domain(ID)明确地获得它。如果 UDF 在一个多相解中需要到一个域指示器的途径，那么函数有必要把适当相的标识符传送到 Get_Domain。

- 例子

下面的这个名为 execut_at_end 的 UDF，使用 DEFINE_EXECUTE_AT_END 对湍流耗散在整个域上进行积分，并且在当前迭代或时间步的最后于控制台窗口输出 UDF 的值。在 FLUENT 中，这个 UDF 可以作为一个通译的或汇编的 UDF 被执行。

```
/*******************************************************************
  UDF：对湍流耗散进行积分，在当前迭代的或时间步的最后于控制台窗口输出所得积分值
*******************************************************************/
    #include “udf.h”

    DEFINE_EXECUTE_AT_END(execute_at_end)
    {
      Domain *d;
     Thread *t;
      /*对耗散进行积分. */
      real sum_diss=0.;
      cell_t c;
      d = Get_Domain(1);    /*若是多相的，则是混合域*/

      thread_loop_c (t,d)
      {
       if (FLUID_THREAD_P(t))
          {
            begin_c_loop (c,t)
                  sum_diss += C_D(c,t)* C_VOLUME(c,t);
            end_c_loop (c,t)
          }
     }
```

```
    printf("Volume integral of turbulent dissipation:%f\n,sum_diss);
    fflush(stdout);
  }
```

3. DEFINE_INIT

使用 DEFINE_INIT 宏可以为解定义一套初始值。DEFINE_INIT 作为补充可以完成相同的结果，只不过是借助一个 UDF 使用一个不同的方法实现的。这个 DEFINE_INIT 函数初始化一次被执行一次，并且在求解器执行了缺省的初始化之后，函数被立即调用。由于流场初始化后函数才被调用，比较有代表性的是它被用来设置流量的初始值。

- 应用

宏：	DEFINE_ADJUST (name, d)
自变量类型：	Domain *d
函数返回：	void

DEFINE_AD_JUST 中有两个自变量：name 和 d。name 是由用户给出的 UDF 的名字。传送的自变量，d，是域的一个指示器，其中初始化函数就会在这个域上被使用。域自变量提供到网格中所有单元的和面的网络之途径。对于多相流动，通过求解器传送到这个函数的域指示器是平均混合物的域指示器。

- 例子

下面的这个名为 my_init_func 的 UDF，初始化一个解中的流场。它在这个解处理的开始被执行一次。在 FLUENT 中，这个 UDF 可以作为一个通译的或汇编的 UDF 被执行。

```
/*****************************************************************
  UDF：初始化流场变量
*****************************************************************/
   #include "udf.h"

   DEFINE_INIT(my_init_func, domain)
   {
       cell_t c;
       Thread *t;
     real xc[ND_ND]

     /*轮循于域中的所有单元脉络 */
     thread_loop_c (t,domain)
         {
             /*轮循于所有的单元*/
             begin_c_loop_all (c,t)
               {
                 C_CENTROID(xc,ct);
                 If (sqrt(ND_SUM(pow(xc[0] – 0.5,2.),
                                 pow(xc[1] – 0.5,2.),
                                 pow(xc[2] – 0.5,2.)))<0.25)
                   C_T(c,t) = 400.;
```

```
            else
               C_T(c,t) = 300.;
          }
        end_c_loop_all (c,t)
      }
  }
```

宏 ND_SUM(a,b,c)计算前两个自变量（2D）或三个自变量（3D）的和。这对于写关于涉及到矢量操作的函数，以便同样的函数可以用于 2D 和 3D 是非常有用的。对于 2D 的情况，不考虑第三个变量。

4. DEFINE_ON_DEMAND

使用 DEFINE_ON_DEMAND 宏可以定义一个按命令执行的 UDF，这比让 FLUENT 在计算过程中机械地调用 UDF 优越的多。UDF 一旦被激活，可以立即被执行，不过，当求解器正在迭代的时候，函数是不可以调用的。注意，域指示器 d 作为 DEFINE_ON_DEMAND 的一个自变量，它的传送是明确的。所以，如果想在这个 UDF 中使用域变量，那么首先需要使用 FLUENT 提供的 Get_Domain 工具重新得到它。

- 应用

宏：	DEFINE_ON_DEMAND (name)
自变量类型：	none
函数返回：	void

DEFINE_ON_DEMAND 中仅有一个自变量：name。name 是由用户给出的 UDF 的名字。

- 例子

下面的这个名为 on_demand_calc 的 UDF，计算和输出关于当前数据场的温度最小值，最大值和平均值。然后，再计算一个温度函数

$$f(T)=\frac{T-T_{\min}}{T_{\max}-T_{\min}}$$

并且把它存放到自定义内存位置 0。一旦连接了这个 UDF，关于 $f(T)$ 的这个场值将在 FLUENT 的后处理面板中的下拉菜单中变地可用。用户可以在自定义内存…类中通过选择 udm-0 选择这个场。如果在执行 UDF 之后写了一个数据文件，那么这个自定义内存场将会保存到这个文件中。相应的源代码可以在 FLUENT 中被通译或汇编。

```
/*********************************************************************
  UDF：计算温度场的函数并存放在自定义内存中。输出温度的最小值，最下值和平均值。
*********************************************************************/
    #include “udf.h”

    DEFINE_ON_DEMAND(on_demand_calc)
    {
     Domain *d;   /*自从作为一个到 DEFINE 宏的自变量，传布域指示器*/
     real trag = 0.;
     real tmax = 0.;
     real tmin = 0.;
     real temp,volume,vol_tot;
```

```
    Thread*t:
    cell_t c;
    d = Get_Domain(1);    /*使用 FLUENT 设备获得域*/
    /*轮循于域中的所有单元脉络*/
    thread_loop_c(t,d)
     {
      /*计算温度的最大值，最小值和体积平均值*/
      /*轮循于所有的单元*/
      begin_c_loop(c,t)
        {
          volume = C_VOLUME(c,t);    /*获得单元体积*/
          temp = C_T(c,t);           /*获得单元温度*/

          if (temp < tmin || tmin == 0.) tmin = temp;
          if (temp > tmax || tmax == 0.) tmax = temp;

          vol_tot += volume;
          tavg += temp*volume;
        }
      end_c_loop(c,t)

      tavg /= vol_tot;

      printf("\n Tmin = %g  Tmax = %g\n  Tvag = %g\n",tmin,tmax,tavg);
      /*计算温度函数，并存放在自定义内存中*/
      /*(位置指标 0)                    */

      begin_c_loop(c,t)
          {
              temp = C_T(c,t);
              C_UDMI(c,t,0) = (temp-tmin)/(tmax-tmin);
           }
      end_c_loop(c,t)
      }
  }
```

名为 on_demand_calc 的函数没有任何明确的自变量。在函数体内，首先函数所用的变量被定义和初始化。在宣布了函数变量之后，一个轮循宏被用来轮循过域中的每一个单元，一个轮循到另一个轮循，从而轮循过所有的单元。在轮循内部，总体积以及温度的最大值，最小值和体积上平均值得到了计算。这些计算结果被输出到 FLUENT 的控制台。然后过每一个单元的第二次轮循用来计算函数 $f(T)$，并把这个函数存放在自定义内存的 0 位置。

5. DEFINE_RW_FILE

使用 DEFINE_RW_FILE 宏可以定义要写进或者读入一个 case 或 data 文件的用户化信息。使用 DEFINE_RW_FILE 可以保存或重存储任何 data 类型（例如，整数，实数，布尔数学体系，结构等类型）的自定义变量。当解正在被计算时，保存相应的动态信息往往是很有用的，这也是此函数的另一种应用。注意，在使用这个函数时，读的顺序和写的顺序必须一样。

- 应用

宏：	DEFINE_RW_FILE (name, fp)
自变量类型：	FILE *fp
函数返回：	void

DEFINE_ON_DEMAND 中有两个自变量：name 和 fp。name 是由用户给出的 UDF 的名字。被传送的自变量，fp，是一个关于正在写进或读入的文件的指示器。

！不要在 DEFINE_RW_FILE 函数里使用正运行于 Windows 平台的 fwrite 宏，而是使用 fprintf。

- 例子

下面的 C 源代码列表包含了关于函数把信息写到一个 data 文件以及又读出的例子。这些函数被连接进一个单独的源文件，其中这个文件可以在 FLUENT 中被通译或编译。

```
/******************************************************************
  UDF：增加一个变量，把它写进一个 data 文件以及把它再从文件中读入 FLUENT。
******************************************************************/
   #include “udf.h”

   int kount =0;   /*定义全局变量 kount*/
   DEFINE_ADJUST(demo_calc,domain)
   {
     kount++
     printf(“kount = %d\n”,kount);
   }
   DEFINE_RW_FILE(writer, fp)
   {
     printf(“Writing UDF data to data file…\n”);
     fprintf(fp,”%d”,kount);    /*写 kount 到 data 文件*/
   }
   DEFINE_RW_FILE(reader, fp)
   {
     printf(“Reading UDF data from data file…\n”);
     fscanf(fp,”%d”,&kount);    /*从 data 文件中读 kount*/
   }
```

一开始，定义了整型变量 kount，给出其初始值 0。第一个函数（demo_calc）是一个 ADJUST 函数，自从每一迭代 ADJUST 函数被调用一次以来，这个函数在每一个迭代上都会增加 kount 的值。当保存 data 文件时，第二个函数（writer）指示 FLUENT 把当前的 kount 值写进 data 文件中。当读入 data 文件时，第三个函数（reader）指导 FLUENT 从 data 文件中读出存储过的 kount

值。

与此同时，如果用户在运行自己的计算，比如说 10 个迭代，保存 data 文件，那么当前的 kount 的值（10）将被保存到这个 data 文件中。如果把这个 data 文件读入 FLUENT 中，让它继续计算，那么 kount 将从 10 这个值开始计算，计算中每一迭代被增加一次。注意，用户可以按自己的意愿保存尽可能多的静态变量，但是必须保证读它们的顺序一定要与它们被写入的顺序一致。

18.2.3: 物种模型的 DEFINE 宏

物种模型的 DEFINE 宏是用来为 FLUENT 中某个特殊的模型设置参数的。表 18.2.3.1 给出了一个关于 DEFINE 宏、DEFINE 宏定义的函数、以及 FLUENT 中被激活的面板的快捷参考指导。每一个 DEFINE 宏的定义都被放在了 udf.h 头文件中。为方便读者，把它们列在了附录 A 中。

此模型主要包括以下几种特殊的 DEFINE 宏：

- DEFINE_CHEM_STEP
- DEFINE_DELTAT
- DEFINE_DIFFUSIVITY
- DEFINE_DOM_DIFFUSE_REFLECTIVITY
- DEFINE_DOM_SOURCE
- DEFINE_DOM_SPECULAR_REFLECTIVITY
- DEFINE_HEAT_FLUX
- DEFINE_NET_REACTION_RATE
- DEFINE_NOX_RATE
- DEFINE_PR_RATE
- DEFINE_PRANDTL
- DEFINE_PROFILE
- DEFINE_PROPERTY
- DEFINE_SCAT_PHASE_FUNC
- DEFINE_SOURCE
- DEFINE_SR_RATE
- DEFINE_TURB_PREMIX_SOURCE
- DEFINE_TURBULENT_VISCOSITY
- DEFINE_UDS_FLUX
- DEFINE_UDS_UNSTEADY
- DEFINE_VR_RATE

表 18.2.3.1：关于物种模型的 DEFINE 宏的快捷参考指导

函数	DEFINE 宏	激活的面板
物种的质量组份	DEFINE_PROFILE	boundary condition（例如，速度进口）
体积组份（多相模型）	DEFINE_PROFILE	boundary conditon
一边界上的速度	DEFINE_PROFILE	boundary conditon
一边界上的压力	DEFINE_PROFILE	boundary conditon
边界温度	DEFINE_PROFILE	boundary conditon
湍流动能	DEFINE_PROFILE	boundary conditon

湍流耗散率	DEFINE_PROFILE	boundary conditon
壁面热条件	DEFINE_PROFILE	boundary conditon
壁面剪切应力	DEFINE_PROFILE	boundary conditon
质量源	DEFINE_SOURCE	boundary conditon
动量源	DEFINE_SOURCE	boundary conditon
能量源	DEFINE_SOURCE	boundary conditon
湍流 k.e.源	DEFINE_SOURCE	boundary conditon
湍流耗散率源	DEFINE_SOURCE	boundary conditon
热通量	DEFINE_HEAT_FLUX	User_Defined Function Hooks
湍流粘性	DEFINE_TURBULENT_VISCOSITY	Viscous Model
密度	DEFINE_PROPERTY	Materials
粘性	DEFINE_PROPERTY	Materials
质量扩散率	DEFINE_PROPERTY	Materials
热传导率	DEFINE_PROPERTY	Materials
吸收系数	DEFINE_PROPERTY	Materials
散射系数	DEFINE_PROPERTY	Materials
层流流动速度	DEFINE_PROPERTY	Materials
应变率	DEFINE_PROPERTY	Materials
物种或 UDS 扩散率	DEFINE_DIFFUSIVITY	Materials
散射相函数	DEFINE_SCAT_PHASE_FUNC	Materials
表面反应率	DEFINE_SR_RATE	Materials
体积反应率	DEFINE_VR_RATE	User-Defined Function Hooks
粒子反应率	DEFINE_PR_RATE	User-Defined Function Hooks
物种净质量反应率	DEFINE_NET_REACTION_RATE	User-Defined Function Hooks

物种净质量反应率在时间步上的积分	DEFINE_CHEM_STEP	User-Defined Function Hooks
湍流与混合料源	DEFINE_TURB_PREMIX_SOURCE	User-Defined Function Hooks
标量通量函数	DEFINE_UDS_FLUX	User-Defined Scalars
标量非稳态函数	DEFINE_UDS_UNSTEADY	User-Defined Scalars
NOx 形成率	DEFINE_NOX_RATE	NOx Model
时间步（关于依赖时间的解）	DEFINE_DELTAT	Iterate
普及的反射率（关于离散纵坐标（DO）模型）	DEFINE_DOM_DIFFUSE_REFLECTIVITY	User_Defined Function Hooks
特殊的反射率（关于 DO 模型）	DEFINE_DOM_DIFFUSE_REFLECTIVITY	User_Defined Function Hooks
源（关于 DO 模型）	DEFINE_DOM_SOURCE	User_Defined Function Hooks
Prandtl 数	DEFINE_PRANDTL	Viscous Models
关于与混合材料的自定义混合法则（密度，粘性系数，传导率）	DEFINE_PROPERTY	Materials
关于 VOF 多相模型的表面张力系数	DEFINE_PROPERTY	Phase Interaction

1. DEFINE_CHEM_STEP

使用 DEFINE_CHEM_STEP 宏可以通过在所给时间步上的积分获得所有物种的净质量反应率：

$$Y_i^{\Delta t} = Y_i^o + \int_0^{\Delta t} \frac{dY_i}{dt} dt \qquad (18.2.3.1)$$

其中 Y_i^0 是物种 i（第 i 种物质）的初始质量组份，t 是时间，Δt 是所给的时间步。$\frac{dY_i}{dt}$ 是净质量反应率。$Y_i^{\Delta t}$ 是第 i 种物质在最后的总质量种所占的质量组份。

DEFINE_CHEM_STEP UDFs 用于 EDC 和 PDF 的输运模型。

- 应用

宏：	DEFINE_CHEM_STEP (name, ifail, n, dt, p, temp, yk)
自变量类型：	int *ifail
	int n
	real dt

real *p

real *temp

real *yk

函数返回：　void

DEFINE_CHEM_STEP 中有七个自变量：name，ifail，n，dt，p，temp，和 yk。name 是用户所给出的 UDF 的名字。，ifail，n，dt，p，temp，和 yk 是被 FLUENT 的求解器传送到 UDF 的变量。如果积分成功，被传送的变量，ifail，是一个被设置为 0 的标记，否则，返回到一个非 0 的值。n 是容积物种数。dt 是时间步，p 是压力，temp 是温度，yd 是包含物种初始质量组份的一个排列。时间步积分后，函数所输出的是质量组份 yk 列。在 yk 列中初始的质量组份则被覆盖掉。

- 例子

下面名为 user_chem._step 的 UDF，假设净体积反应率的表达式为：

$$\frac{dY_k}{dt} = 1/N_{spe} - Y_k \tag{18.2.3.2}$$

其中，N_{spe} 是物质的种数。

关于这个 UDF 的积分的一个分析解是，

$$Y_k^{\Delta t} = \left(Y_k^0 - 1/N_{spe}\right)\exp(-\Delta t) + 1/N_{spe} \tag{18.2.3.3}$$

/***
关于演示 DEFINE_CHEM_STEP 的 UDF 的例子
***/

```
#include “udf.h”

DEFINE_CHEM_STEP(user_chem_step, ifail,n, dt, p, temp, yk)
{
  int i;
  real c = 1./n_spe, decay = exp(-dt);
  for(i=0;i<spe;i++)
    yk[i] = (yk[i]-c)*decay + c;
}
```

2. DEFINE_DELTAT

可以使用 DEFINE_DELTAT 宏，在求解一个依赖于时间的问题的过程中控制时间步的大小。注意，这个宏当且仅当 FLUENT 中的 **Itetate** 面板里的适应时间步方法已被激活的情况下才可使用。

- 应用

宏：　DEFINE_DELTAT (name,domain)

自变量类型：　Domain *domain

函数返回：　real

DEFINE_DELTAT 中有两个自变量：name 和 domain。name 是用户所给出的 UDF 的名字。domain 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要计算物理时间步的实型值，并把此值返回到求解器。domain 是域的一个指示器，其中时间步控制函数就会在这

个域上被使用。域自变量提供到网格中所有单元的和面的网络之途径。对于多相流动，通过求解器传送到这个函数的域指示器是平均混合物的域指示器。

- 例子

下面名为 mydeltat 的 UDF 是一个单独的函数，它展示了怎样在一个模拟过程中使用 DEFINE_DELTAT 改变时间步的值。首先，CURRENT_TIME 被用来取得模拟中当前的时间步之值（它被安置到变量 flow_time）。然后，对于前 0.5 妙的计算，一个时间步的大小是 0.1。其余时间模拟中的时间步长则被设为 0.2。那么，时间步变量被返回到求解器。

```
/**********************************************************************
改变关于一个依赖于时间的解的时间步长的 UDF
**********************************************************************/
 #include “udf.h”

 DEFINE_DELTAT(mydeltat, domain)
 {
   real time_step;
   real flow_time = CURRENT_TIME;
   if (flow_time < 0.5)
       time_step = 0.1;
   else
       time_step = 0.2;
   return time_step;
 }
```

3. DEFINE_DIFFUSIVITY

使用 DEFINE_DELTAT 宏可以为物种的输运方程或者自定义标量（UDS）输运方程规定扩散率。

- 应用

宏：	DEFINE_DIFFUSIVITY (name, c, t, i)
自变量类型：	cell_t c
	Thread *t
	int i
函数返回：	real

DEFINE_DIFFUSIVITY 中有四个自变量：name，c，t，和 i。name 是用户所给出的 UDF 的名字。c，t，和 i 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要计算仅关于一个单元的扩散率，并返回到求解器里的实型值。

c 是一个标识符，它用来鉴别所给定脉络中的一个单元。t 是脉络的一个指示器，其中扩散率方程应用于此脉络上。i 是鉴别物种或自定义标量的一个指标。

注意，扩散率 UDFs 从一个在单元脉络上的轮循里被 FLUENT 访问。因此， UDF 将不必轮循过一个脉络中的所有单元，因为 FLUENT 在访问函数的时候也在进行此项工作。

- 例子

下面名为 mean-age-diff 的 UDF，使用一个自定义标量计算空气平均龄的扩散率。注意，空气龄的计算不要求执行能量，辐射或物种输运的计算。在模型中，需要在进口和出口处设 uds-0=0.0。这个方程可以作为一个通译的或编译的 UDF 被执行。

```
/*********************************************************************
使用一个自定义标量计算空气平均龄的 UDF
*********************************************************************/
 #include “udf.h”

 DEFINE_DIFFUSIVITY(mean_age_diff, c, t, i)
 {
       return C_R(c,t) * 2.88e-05 + C_MU_EFF(c,t)/0.7;
 }
```

4. DEFINE_DOM_DIFFUSE_REFLECTIVITY

如果打算修改由 FUENT 计算所得的关于广泛反射的半透明壁面上的内表面的反射率，那么可以使用 DEFINE_DOM_DIFFUSE_REFLECTIVITY 宏。在执行过程中，对于每一个半透明壁面和每一波段（在一个非灰的离散纵坐标模型（Discrete Ordinates Model, DOM）的情况下），一个 DEFINE_DOM_DIFFUSE_REFLECTIVITY 函数被 FLUENT 访问。所以，此函数可以用来修改界面上的扩散的反射率和扩散的传播率之值。

- 应用

宏：	DEFINE_DOM_DIFFUSE_REFLECTIVITY (name, t, nb, n_a, n_b, diff_ref_a, diff_tran_a, diff_ref_b, diff_tran_b)
自变量类型：	Thread *t
	int nb
	real n_a
	real n_b
	real *diff_ref_a
	real *diff_tran_a
	real *diff_ref_b
	real *diff_tran_b
函数返回：	void

DEFINE_DOM_DIFFUSE_REFLECTIVITY 中有九个自变量：name，t，nb，n_a，n_b，diff_ref_a，diff_tran_a，diff_ref_b 和 diff_tran_b。name 是用户给出的 UDF 的名字。其它八个自变量是被 FLUENT 的求解器传送到 UDF 的变量。

被传送的变量，t，是脉络的一个指示器，其中离散坐标扩散率方程应用于这个脉络上。nband 表示波段数，这个自变量在离散纵坐标模型是非灰的时候才需要。n_a 和 n_b 表示介质 a 和 b 的折射指数。变量 diff_ref_a 和 diff_tran_a 分别是界面上面对介质 a 的扩散反射率和扩散传播率。类似地，diff_ref_b 和 diff_tran_b 分别是界面上面对介质 b 的扩散反射率和扩散传播率。

- 例子

下面名为 user_chem._step 的 UDF，修改在隔离介质 a 和 b 的界面的两边上的扩散反射率和传播率之值。UDF 调用与半透明壁面，并输出关于边 a 和 b 的扩散反射率和传播率的值。

```
/*UDF，输出半透明壁面上的扩散的反射率和扩散的传播率*/

#include “udf.h”

DEFINE_DOM_DEFFUSE_REFLECTIVITY(user_dom_diff_refl, t, nband, n_a, n_b,
```

```
diff_ref_a, diff_tran_a, diff_ref_b, diff_tran_b)
{
/*Argument types
  Thread *t
  int nband
  int n_a
  int n_b
  real *diff_ref_a    //边 a 上的扩散的反射率
  real *diff_tran_a  //边 a 上的扩散的传播率
  real *diff_ref_b   //边 b 上的扩散的反射率
  real *diff_tran_b  //边 b 上的扩散的传播率
 */
printf("diff_ref_a=%f diff_tran_a=%f \n", *diff_ref_a, *diff_tran_a);
printf("diff_ref_b=%f diff_tran_b=%f \n", *diff_ref_b, *diff_tran_b);
}
```

5. DEFINE_DOM_SOURCE

可以使用 DEFINE_DOM_SOURE 宏改变关于离散的纵坐标（Discrete Ordinates，（DO））模型的辐射输运方程中的散发项和分散项。

- 应用

宏：	DEFINE_DOM_SOURCE (name, c, t, ni,nb, emission, in_scattering, abs_coeff, scat_coeff)
自变量类型：	cell_t c Thread *t int ni int nb real *emission real *in_scattering real *abs_coeff real *scat_coef
函数返回：	void

DEFINE_DOM_SOURCE 中有九个自变量：name，c，ni，nb，emission，in_scattering，abs_coeff 和 scat_coeff。name 是用户给出的 UDF 的名字。其它八个自变量是被 FLUENT 的求解器传送到 UDF 的变量。

c 是区别单元的一个指标。t 是单元脉络的一个指示器。ni 表示立体角的方向。nb 是波段数，这个自变量在非灰的 DO 模型中才需要。emission 和 in_scattering 分别是辐射输运方程中的散发项和分散项的指示器。abs_sodff 和 scat_coeff 分别是吸收和分散系数的指示器。

- 例子

在下面名为 user_dom_source 的 UDF 中，辐射输运方程中的散发项被做了修改。对于所有的网格，FLUENT 都访问了这个 UDF，并且 UDF 把散发增加了 5%。

```
/*UDF，改变 DO 模型中的散发项*/

#include "udf.h"

DEFINE_DOM_SOURCE(user_dom_source,c,t,ni,nb,emission,in_scattering,
abs_coeff, scat_coeff)
```

```
{
  /*散发增加 5%*/
  *emission *= 1.05;
}
```

6. DEFINE_DOM_SPECULAR_REFLECTIVITY

可以使用 DEFINE_DOM_SPECULAR_RE 宏改变特殊地反射半透明壁面的内表面的反射率。如果反射率取决于标准边界条件所不允许的其它条件，也许用户希望如此做。在 FLUENT 执行过程中，关于半透明壁面的所有面，对于每一用法，同一个 UDF 被调用。

- 应用

宏：	DEFINE_DOM_SPECULAR_REFLECTIVITY (name, f, t, nband,n_a,n_b,ray_direction, en,internal_reflection, specular_reflectivity,specular_transmissivity)
自变量类型：	face_t f Thread *t int nband real n_a real n_b real ray_direction real en int internal_reflection real *specular_reflectivity real *specular_transmissivity
函数返回：	void

DEFINE_DOM_SPECULAR_REFLECTIVITY 中有十一个自变量：name，f，t，nband，n_a，n_b，ray_direction，en，internl_reflection，specular_reflectivity，和 specular_transmissivity。name 是用户给出的 UDF 的名字。其它十个自变量是被 FLUENT 的求解器传送到 UDF 的变量。

被传送的变量，f，是一个鉴别面的指标，t 是面脉络的一个指示器，其中镜子的反射率函数应用于此脉络上。nband 表示波段数，这个自变量在非灰的 DO 模型中才需要。n_a 和 n_b 分别表示介质 a 和 b 的折射指数。ray_direction 表示方向矢量。en 是界面法线。internal_reflectiony 用来标记所发生的总的反射体的代码。specular_reflectivity 和 specular_transmissivity 分别是关于所给定方向 s 的镜子反射率和镜子传播率。

- 例子

在下面名为 user_dom_spec_refl 的 UDF 中，对于一个给定的面 f 上的光线方向 s，镜子的反射率和传播率被做了改变。

```
/*UDF，在半透明的壁面上，沿面 f 的方向改变镜子的反射率和传播率*/

#include “udf.h”

DEFINE_DOM_SPECULAR_REFLECTIVITY(user_dom_spec_refl, f, t, nband, n_a,
n_b, ray_direction, en, internal_reflection,
specular_reflectivity,specular_transmissivity)
{
/*Argument types
  face_t f
```

```
    Thread *t
    int_nband
    int n_a   (介质 a 的折射指数)
    int n_b   (介质 b 的折射指数)
    real ray_direction
    real ray_direction
    real en
    int*internal_reflection
      real *specular_reflectivity
      real *specular_transmissivity
  */
    real angle, cos_theta;
    real PI = 3.141592;
    cos_theta = NV_DOT(ray_direction, en);
    if (angle >45  &&angle < 60)
      {
        *specular_reflectivity = 0.3;
        *specular_transmissivity = 0.7;
      }
  }
```

7. DEFINE_HEAT_FLUX

不管 DEFINE_HEAT_FLUX 宏的名字是什么，DEFINE_HEAT_FLUX 宏不被用来明确地设置出沿一个壁面的热通量。当打算使用其它的、当前还没有建立模型的热传递机制时，必须使用一个 DEFINE_HEAT_FLUX UDF。壁面上总的热通量将是当前计算的热通量之和，且热通量是由 UDF 定义的。

- 应用

宏：	DEFINE_HEAT_FLUX (name, f, t, co, to, cid, cir)
自变量类型：	face_t f
	Thread *t
	cell_t co
	Thread *t0
	real cid[]
	real cir[]
函数返回：	void

DEFINE_HEAT_FLUX 中有七个自变量：name，f，t，c0，t0，cid 和 cir。name 是用户给出的 UDF 的名字。f，t，c0，t0，cid 和 cir 是被 FLUENT 的求解器传送到 UDF 的变量。

被传送的变量，f，是一个指标，它被用来鉴别所给定脉络中的一壁面的面的。t 是面脉络的一个指示器，其中热通量函数应用于此脉络上。c0 是一个识别邻近壁面的这个单元的指标，t0 是邻近单元的脉络的一个指示器。

cid[]和 cir[]是 UDF 计算所需要的实型排列。排列 cid[]中存储了流体边的扩散热传递系数，排列 cir[]中存储了辐射热传递系数。随着它们的输出被代入到函数，FLUENT 根据以下方程计算辐射热通量（qid）和辐射热通量（qir）：

qid=cid[0] + cid[1]*C_T(c0, t0) – cid[2]*F_T(f, t) – cid[3]*pow(F_T(f, t), 4)

qir=cir[0] + cir[1]*C_T(c0, t0) – cid[2]*F_T(f, t) – cid[3]*pow(F_T(f, t), 4)

qir 和 qid 的和定义了从流体到壁面（这个为正方向）的总热通量，并且从壁面的一个

能量平衡角度讲，它等于周围即从外部到域的热通量。注意，热通量 UDFs 从一个轮循于壁面的所有面内开始被 FLUENT 访问。

！考虑到求解器对 C_T 和 F_T 计算，所给 cid[1]和 cid[2]的值不应当为 0。

8. DEFINE_NET_REACTION_RATE

可以使用 DEFINE_NET_REACTION_RATE 宏来返回所有物种的质量净反应率。一种物质的质量净反应率等于所有反应的体积反应率的总和与这种物质的分子重量的乘积：

$$\frac{dY_i}{dt} = M_i \sum_{r=1}^{N_R} \hat{R}_r \qquad (18.2.3.4)$$

其中，Y_i 是物种 i 的质量组份，t 是时间，M_i 是物种 i 的分子重量，$\hat{R}_r$ 是第 r 个反应的摩尔反应率，N_R 是总的反应个数。一个 DEFINE_NET_REACTION_RATE UDF 不仅对在表面化学的模型有所应用，而且也可以用于 EDC 和 PDF 输运模型。相反的，体积 UDF 函数 DEFINE_VR_RATE 和表面 UDF 函数 DEFINE_SR_RATE 在每一反应中都会返回摩尔率。也许，用户想使用 DEFINE_NET_REACTION_RATE 连接一个提供反应率的外部程序，例如 CHEMKIN。

- 应用

宏：	DEFINE_NET_REACTION_RATE (name, p, temp, yi, rr, jac)
自变量类型：	double *p
	double *temp
	double *yi
	double *rr
	double *jac
函数返回：	void

DEFINE_NET_REACTION_RATE 中有六个自变量：name，p，temp，yi，rr 和 jac。name 是用户给出的 UDF 的名字。p，temp，yi，rr 和 jac 是被 FLUENT 的求解器传送到 UDF 的变量。

输入的变量有压力 p，温度 temp，和物种的质量组份的排列 yi。函数输出的变量是质量净反应率的排列，rr，以及 Jacobian 排列 jac。注意，仅在表面化学中需要 Jacobian，是一个指标，Jacobian 是关于物种浓度的表面净质量反应率的导数。

- 例子

下面名为 user_net_reaction_rate 的 UDF 假设净体积反应率的表达式为：

$$\frac{dY_i}{dt} = 1/N_{spe} - Y_i \qquad (18.2.3.5)$$

其中，N_{spe} 是物种个数。

```
/*********************************************************************
演示 DEFINE_NET_REACTION_RATE 的 UDF
*********************************************************************/
    #include “udf.h”

    DEFINE_NET_REACTION_RATE(user_net_reaction_rate, P, temp, yi,
```

```
rr,jac)
{
    int I;
    for(I=;I<n_spe;I++)
      rr[I] = 1./(real)n_spe – yi[I];
}
```

9. DEFINE_NOX_RATE

在 FLUENT 中，可以使用 DEFINE_NOX_RATE 宏计算 NOx 的生成率和减少率。所规定的 UDF 率不受标准的 NOx 模型选项的约束。可以在模拟中取消选择标准的 NOx 模型选项，而选择自定义函数的速率。

- 应用

宏：	DEFINE_NOX_RATE (name, c, t, NOx)
自变量类型：	cell-t c
	Thread *t
	NOx_Parameter *yi
函数返回：	void

DEFINE_NOX_RATE 中有四个自变量：name，c，t 和 NOx。name 是用户给出的 UDF 的名字。c，t 和 NOx 是被 FLUENT 的求解器传送到 UDF 的变量。

c 是一个指数，它被用来区别所给定脉络中的一个单元的。t 是脉络的一个指示器，其中 NOx 率将应用于此脉络上。变量 NOx 是 NOx 结构的一个指示器。一个 DEFINE_NOX_RATE 函数不返回到一个值，所计算的 NOx 率是通过 NOx 结构被返回的。

！注意，虽然数据结构被称作 NOx，但是 DEFINE_NOX_RATE 宏可被用来计算任何污染物种（例如，NO，HCN 和 NG_3）的速率，这取决于所求解的是哪些污染物种之方程。

- 例子

下面名为 user_nox 的 UDF 计算 NOx 的生成率和减少率，根据 NO 的前进和后退速度它们被定义如下：

$$R_f = \frac{2[O]k_1k_2[O_2][N_2]}{k_2[O_2]+k_{-1}[NO]} \qquad (18.2.3.6)$$

和

$$R_r = \frac{2[O]k_{-1}k_{-2}[NO]^2}{k_2[O_2]+k_{-1}[NO]} \qquad (18.2.3.7)$$

其中，速率系数的单位为 $m^3/(mol\cdot s)$，并且它们被定义为：

$$k_1 = 1.8\times10^8 \exp\left(\frac{-38370}{T}\right) \qquad (18.2.3.8)$$

$$k_{-1} = 3.8\times10^7 \exp\left(\frac{-425}{T}\right) \qquad (18.3.3.9)$$

$$k_2 = 1.8 \times 10^4 \exp\left(\frac{-4680}{T}\right) \qquad (18.2.3.10)$$

$$k_{-2} = 3.8 \times 10^4 \exp\left(\frac{-20820}{T}\right) \qquad (18.2.3.11)$$

O 的浓度的表达式为：

$$[O] = 3.664 \times 10^1 T^{0.5} [O_2]^{0.5} \exp\left(\frac{-27123}{T}\right) \qquad (18.2.3.12)$$

在速率表达式中，所有物质的浓度单位是 mol/m^3 。

```
/***********************************************************************
关于自定义 NOx 速率的 UDF 例子
***********************************************************************/
    #include “udf.h”

    #difine SMALL_S 1.e-29

    DEFINE_NOX_RATE(user_nox, c, t, NOx)
    {
        real kf1, kr1, kf2, kr2;
        real o_eq;
        real s1, s2, s3, rf, rr;
        Rate_Const K_F[2] = {{1.80e8, 0.0, 38370.0},
                             {1.80e4, 1.0, 4680.0}};
        Rate_Const K_R[2] = {{3.80e7, 0.0, 425.0},
                             {3.80e3, 1.0, 20820.0}};
        Rate_Const K_0 = {3.664e1, 0.5, 27123.0},

        If (NOX_EQN(NOx) !=EQ_NO) return;

        kf1 = ARRH(NOx, K_F[0]);
        kr1 = ARRH(NOx, K_R[0]);
        kf2 = ARRH(NOx, K_F[1]);
        kr2 = ARRH(NOx, K_R[1]);

        s1 = kf2*MOLECON(NOx, 02);
        s3 = s1 + krl*MOLECON(NOx, NO);
        /*决定 O 的浓度（局部平衡）*/
        o_eq = ARRH(NOx, K_0)*sqrt(MOLECON(NOx, 02));

        /*计算 NO 速率*/
        s2 = 2.*o_eq;
        /*前进速率…*/
        rf = s2*(kf1*MOLECON(NOx, N2))*s1/s3;
        /*后退速率…*/
        rr = -s2*(kf1*MOLECON(NOx, N2))*s1/s3;
```

```
        /*速率的单位是 mole/m^3/s*/
        NOX_FRATE(NOx) = rf;
        NOX_RRATE(NOx) = rr;
    }
```

使用 UDFs 可以把 FLUENT 所提供的一些宏用在计算污染物质速率中。下面所列出的宏定义在头文件 sg_nox.h 里，而此文件又被包含在 udf.h 文件里。变量 NOx 表示 NOx_Parameter 结构的一个指示器。

- NOX_EQN(NOx)返回到当前所求解的污染物方程的指数。这些指数是关于 NO 的 EQ_NO，关于 HCN 的 EQ_HCN，以及关于 NH_3 的 EQ_NH3。
- MOLECON(NOx，SPE)返回到由 SPE 所规定的一种物质的摩尔浓度，并且 SPE 必须由下面的标识符中的某个所取代：FUEL，O_2，O，OH，H_2O，N_2，N，CH，CH_2，CH_3，NO，HCH，NH_3。标识符 FUEL 表示物种燃料，它是在 **NOx Model** 面板中的 **Prompt NO Parameters** 之内的 **Fuel Species** 所对应的列表中所规定的。
- 如果由 SPE 所规定的物种在 FLUENT 对应的条件（例如，在 **Species** 面板）下不存在，那么 NULLIDX(NOx，SPE)返回到 TRUE。
- ARRH(NOx，K)返回值是 Arrhenius 率，它是从 K 规定的常数计算得到的。K 是使用 Rate_Const 数据类型而定义的，它有三个元素—A，B 和 C。Arrhenius 率定义为：

$$R = AT^{B}\exp(-C/T)$$

  其中，T 是温度。
- NOX_FRATE(NOx)用来返回到所求解的污染物种的生成率。
- NOX_RRATE(NOx)用来返回到所求解的污染物种的减少率。

10. DEFINE_PR_RATE

可以使用DEFINE_PR_RATE宏为多相表面反应粒子模型定义一个自定义粒子表面反应的函数。在 FLUENT 执行过程中，对于所有的粒子表面反应，相同的 UDF 被访问，所以 DEFINE_PR_RATE 可以被用来定义一个单独反应或多相反应的自定义反应率。在耦合求解器被应用时，一个 DEFINE_PR_RATE UDF 不能被访问，所以若使用这个 UDF，则需要使 **Discrete Phase Mode**l 面板中的 **Coupled Heat Mass Solution** 选项不具有功能。下面所给的辅助函数，**zbrent_pr_rate**，可以在全部粒子反应率没有解析解时使用。

- 应用

宏：	DEFINE_PR_RATE (name, c, t, r, mw, ci, p, sf, dif_index, cat_index, rr)
自变量类型：	cell_t c
	Thread *t
	Reaction *r
	real *mw
	real *ci
	Tracked_Particle *p
	real *sf
	int dif_index
	int cat_index
	real *rr
函数返回：	void

DEFINE_PR_RATE 中有十一个自变量：name，c，t，r，mw，ci，p，sf，dif_index，cat_index 和 rr。name 是用户给出的 UDF 的名字。c，t，r，mw，ci，p，sf，dif_index，cat_index 和 rr 是被 FLUENT 的求解器传送到 UDF 的变量。

c 是一个指标，它被用来鉴别粒子当前所包含的单元的，t 是粒子脉络的一个指示器。R 是数据结构的一个指示器，其中这些数据结构表示当前的反应。mw 是是气态的和表面物种分子重量的一个实型指示器的排列。pp 是气体局部压力的一个实型指示器的排列。p 是 Tracked_Particle 数据结构的指示器，其中这些数据结构包含与示踪粒子相联系的数据。sf 是当前时间步的粒子 char 质量中的包含其质量组份的一个实型指示器排列。注意，在排列 sf 中存储的固体物种的质量组份的顺序与 **Materials** 面板里的 **Selected Solid Species** 列表中定义的物种顺序是一致的，其中这些物种是从 **Mixture Material** 的 **Edit Species** 名称选项中打开的。dif_I 和 cat_I 是整型变量，它们是就当前反应而言的，定义在 **Reaction** 面板中的扩散控制物种和催化剂物种， UDF 将需要设置单位为 $kgmol/s$ 的，粒子反应率的实型指示器 rr 的参考变量。

- 一个辅助函数

  zbrent_pr_rate (real (*func),(real,real [],int [],Boolean [],char *,) real ruser[],int iuser[], boolean buser[], char *cuser,real x1 real x2, real tol, boolean *ifail)

  辅助函数返回值：real

- 例子

下面名为 user_pr_rate 的 UDF，规定下面表达式（18.2.3.13）中的一个粒子的反应率 η_r，

$$\overline{R}_{j,r} = A_P \eta_r Y_j R_{kin,r} \qquad (18.2.3.13)$$

其中，$\overline{R}_{j,r}$ 是粒子表面物种耗损率（kg/s），A_P 是粒子表面面积（m^2），Y_j 是粒子中表面第 j 种物种的质量组份，$R_{kin,r}$ 是第 r 个反应的运动率（单位是变化的），η_r 是有效系数即粒子的反应率（无量纲），其表达式为：

$$\eta_r = 1 - x$$

这里，x 是粒子烧焦质量的转化部分。在这种情况下，获得全部表面反应的一个数值解是必要的。

UDF 仅在第二个反应中被调用，这意味着其余的粒子表面反应将使用 FLUENT 的默认解。

```
/***********************************************************************
UDF：应用一个数值解指定一个粒子的表面反应率
***********************************************************************/
    #include “udf.h”

    #define c1 5e-12
    #define A1 0.002
    #define E1 7.9e7
    #define tolerance 1e-4
    #define order 0.9
```

```
real reaction_rate(real rate, real ruser[], int iuser[], boolean
  buser[], char*cuser)
{
    return (ruser[2]*pow(MAX(0.,(ruser[0]-rate/ruser[1])),order)
    –rate);
}
DEFINE_PR_RATE(user_rate, c, t, r, mw, pp, p sf, dif_i, cat_i, rr)
{
  if (!strcmp(r->name,”reaction-2”))
    {
    boolean ifail=FALSE;

    real ash_mass=
    P_INIT_MASS(p)*(1.-DPM_CHAR_FRACTION(p)-DPM_VOLATILE_FRACTION
    (p));

    real one_minus_conv =
    MAX(0.,(P_MASS(p) –ash_mass) / P_INIT_MASS(p)/
    DPM_CHAR_FRACTION(p));
    real ruser[3];
    int iuser[1];
    boolean buser[1];
    char cuser[30];

    real ratemin, ratemax, root;

    ruser[0] = PP[dif_i];
    ruser[1] = MAX(1.E-15,
    (c1*pow(0.5*(P_T(p)+C_T(c,t)),0.75)/P_DIAM(p)));
    ruser[2] = A1*exp(-E1/UNIVERSAL_GAS_CONSTANT/P_T(p));
    strcpy(cuser, “reaction-2”);

    ratemin=0;
    ratemax=ruser[1]*pp[dif_i];
```

/*关于辅助函数 zbrent_pr_rate 的自变量
在这个例子中，在 UDF 的开始定义了一个实型的函数 reaction_rate。reaction_rate 的自变量是 rate 和指示器排列 real ruser[],integer iuser[],boolean buser[]和 char *cuser,它们必须在 DEFINE_PR_RATE 函数的主体中给出宣布和定义。
比较典型的是，如果粒子表面反应率表述为 rate = f(ruser[],iuser[],rate)，那么本例题中的实型方程 reaction_rate 应当返回到 f(ruser[],iuser[],rate) – rate。在定义的率比较复杂的情况下，变量 boolean buser[]和 char *cuser 可以用来控制程序的流动。
ratemin 和 ratemax 分别代表了变量率的最小可能值和最大可能值。它们定义搜寻间隔，其中数值运算法则将在此间隔里寻找定义在函数 reaction_rate 中的方程的根。反应率 rr 的值在公差 tol 的值到达一个准确值时才被改进。
如果方程的根没有找到，那么变量 ifail 的取值将是 TURE。*/

```
    root = zbrent_pr_rate(reaction_rate, ruser, iuser, buser,
```

```
        cuser,ratemin, ratemax, tolerance, &ifail);
        if (ifail) root=MAX(1.E-15,ruser[1]);
        rr=-root*p_DIAM(p)*P_DIAM(p)*M_PI*sf[0]*ome_minus_conv;
        Message("Fail status%d\n",ifail);
        Message("Reaction rate for reaction%s:%g\n",cuser,*rr);
        }
}
```

11. DEFINE_PRANDTL

下面的宏可以用来在 FLUENT 中规定 Prandtl 数。

DEFINE_PRANDTL_D

可以使用 DEDINE_PRANDTL_D 指定湍流耗散（ε）的 Prandtl 数。

- 应用

宏：	DEFINE_PRANDTL_D (name, c, t)
自变量类型：	cell_t c
	Thread *t
函数返回：	real

DEFINE_PRANTL_D 中有三个自变量：name，c 和 t。name 是用户给出的 UDF 的名字。C 和 t 是被 FLUENT 的求解器传送到 UDF 的变量。关于湍流耗散 Prandtl 数，UDF 将需要返回实型值到求解器。被传送的变量 t 是单元脉络的一个指示器，c 是一个指数，它被用来鉴别 Prandtl 数函数将应用的单元。

DEFINE_PRANDTL_K

可以使用 DEDINE_PRANDTL_K 规定湍流动能（k）的 Prandtl 数。

- 应用

宏：	DEFINE_PRANDTL_K (name, c, t)
自变量类型：	cell_t c
	Thread *t
函数返回：	real

DEFINE_PRANTL_K 中有三个自变量：name，c 和 t。name 是用户给出的 UDF 的名字。c 和 t 是被 FLUENT 的求解器传送到 UDF 的变量。关于湍流动能的 Prandtl 数， UDF 将需要返回实型值到求解器。被传送的变量 t 是单元脉络的一个指示器，c 是一个指数，它被用来鉴别 Prandtl 数函数将应用的单元。

- 例子

下面的 UDF 使用 FLUENT 中被激活的$k-\varepsilon$模型实现 RNG 模型的一个高雷诺数版本。需要三个步骤：

1．在 RNG 模型中设置 Crnu，Cleps 和 C2eps。

2．使用 UDF 计算k和ε的 Prandtl 数。

3．在ε方程中添加$-r$源项。

在 RNG 模型中，k与ε两方程中扩散系数的形式为：

$$(\mu+\mu_t)*\alpha$$

而在标准的$k-\varepsilon$模型中，它的形式是：

$$\mu+\frac{\mu_t}{\Pr}$$

对于新的执行，为了获得与原始的 RNG 模型相同的执行，一个 UDF 需要定义一个表达式为

$$\Pr = \frac{\mu_t}{\left|(\mu + \mu_t) * \alpha - \mu\right|}$$

的 Prandtl 数 $\Pr$ 。

下面这个函数演示了这种应用。注意，源代码必须作为一个编译的 UDF 来执行。

```
#include “udf.h”

DEFINE_PRANDTL_K(user_pr_k, c, t)
{
  real pr_k, alpha;
  real mu = C_MU_L(c,t);
  real mu_t = C_MU_T(c,t);

  alpha = rng_alpha(1.,mu + mu_t,mu);
  pr_k = mu_t/((mu+mu_t)*alpha-mu);
  return pr_k;
}
DEFINE_PRANDTL_D(user_pr_d,c,t)
{
  real pr_d, alpha;
  real mu = C_MU_L(c,t);
  real mu_t = C_MU_T(c,t);

  alpha = rng_alpha(1.,mu + mu_t,mu);
  pr_d = mu_t/((mu+mu_t)*alpha-mu);
  return pr_d;
}
DEFINE_SOURCE(eps_r_source,c,t,Ds,eqn)
{
  real con, source;
  real mu   = C_MU_L(c,t);
  real mu_t  = C_MU_T(c,t);
  real k     = C_K(c,t);
  real d     = C_D(c,t);
  real prod  = C_PRODUCTION(c,t);

  real s = sqrt(prod/(mu+mu_t));
  real eta = s*k/d;
  real eta_0 = 4.38;
  real term = mu_t*s*s*s/(1.0 + 0.012*eta*eta*eta);

  source = -term * (1. –eta/eta_0);
  dS[ean] = -term/d;

  return source;
```

```
}
```

DEFINE_PRANDTL_O

可以使用 DEDINE_PRANDTL_O 规定 $k-\varepsilon$ 湍流模型中的比耗散（ω）的 Prandtl 数。

- 应用

宏：	DEFINE_PRANDTL_O (name, c, t)
自变量类型：	cell_t c
	Thread *t
函数返回：	real

DEFINE_PRANTL_O 中有三个自变量：name，c 和 t。name 是用户给出的 UDF 的名字。c 和 t 是被 FLUENT 的求解器传送到 UDF 的变量。关于比耗散的 Prandtl 数，UDF 将需要返回实型值到求解器。被传送的变量 t 是单元脉络的一个指示器，c 是一个指数，它被用来鉴别 Prandtl 数函数将应用的单元。

- 例子

```
/*规定一个常数比耗散 Prandtl 数*/

#include "udf.h"

DEFINE_PRANDTL_O(user_pr_O, c, t)
{
  real pr_O;
  pr_o = 2.;
  return pr_o;
}
```

DEFINE_PRANDTL_T

可以使用 DEDINE_PRANDTL_T 规定出现在温度方程的扩散项中的 Prandtl 数。

- 应用

宏：	DEFINE_PRANDTL_T (name, c, t)
自变量类型：	cell_t c
	Thread *t
函数返回：	real

DEFINE_PRANTL_T 中有三个自变量：name，c 和 t。name 是用户给出的 UDF 的名字。c 和 t 是被 FLUENT 的求解器传送到 UDF 的变量。关于温度 Prandtl 数，UDF 将需要返回实型值到求解器。被传送的变量 t 是单元脉络的一个指示器，c 是一个指数，它被用来鉴别 Prandtl 数函数将应用的单元。

- 例子

```
/*规定一个常数温度 Prandtl 数*/

#include "udf.h"

DEFINE_PRANDTL_T(user_pr_t, c, t)
{
  real pr_t;
  pr_t = 0.85;
  return pr_t;
```

}

DEFINE_PRANDTL_T_WALL

可以使用 DEDINE_PRANDTL_T_WALL 为热壁面函数规定的 Prandtl 数。

- 应用

宏：	DEFINE_PRANDTL_T_WALL (name, c, t)
自变量类型：	cell_t c
	Thread *t
函数返回：	real

DEFINE_PRANTL_T_WALL 中有三个自变量：name，c 和 t。name 是用户给出的 UDF 的名字。c 和 t 是被 FLUENT 的求解器传送到 UDF 的变量。关于热壁面函数的 Prandtl 数，UDF 返回到求解器的值需要是实型的。被传送的变量 t 是单元脉络的一个指示器，c 是一个指数，它被用来鉴别 Prandtl 数函数将应用的单元。

- 例子

/*规定一个热壁面函数的常值 Prandtl 数*/

```
DEFINE_PRANDTL_O(user_pr_O, c, t)
{
  real pr_t_wall;
  pr_t_wall = 0.85;
  return pr_t_wall;
}
```

12. DEFINE_PROFILE

可以使用 DEFINE_PROFILE 宏定义一个自定义边界截面，并且这个边界截面随着一个空间坐标或时间坐标的函数变化。可以在边界上用户化的一些变量为：

- 速度，压力，温度，湍流动能，湍流耗散率
- 物种的质量组份（物种的输运）
- 体积组份（多相）
- 壁面热条件（温度，热通量，热产生率，热传递系数，外部发射率，等等）
- 等等

值得注意的是，对于壁面热通量，DEFINE_PROFILE 只允许修改其中一个单独的值。这个单一值用在 FLUENT 所不能线性化的显示源项中。如果想把关于壁面热通量的源项线性化，并考虑到了传导的和辐射的热传递，那么将需要使用 DEFINE_HEAT_FLUX 规定 UDF。

- 应用

宏：	DEFINE_PROFILE (name, t, i)
自变量类型：	Thread *t
	int i
函数返回：	void

DEFINE_PROFILE 中有三个自变量：name，t 和 i。name 是用户给出的 UDF 的名字。t 和 i 是被 FLUENT 的求解器传送到 UDF 的变量。

被传送的变量，t，是脉络的一个指示器，其中边界条件在这个脉络上应用。i 是一个用来鉴别将被定义的变量的指数。当通过图形用户界面把 UDF 与一个边界条件面板里的一个变量连接起来的时候，设置 i。

DEFINE_PROFILE 通常用来规定一个边界面区域上的一个剖面条件，另一方面，它也可以用来规定或固定流动变量，在计算过程中，这些流动变量在一个单元域里取常数。

注意，与源项和特征 UDFs 不同，剖面 UDFs（使用 DEFINE_PROFILE 定义的）不会从边界区域的脉络上的一个轮循里被 FLUENT 访问。求解器仅把指示器传送到与 DEFINE_PROFILE 宏的边界区域相关的脉络。UDF 将需要轮循过脉络中的所有面，计算有关这个边界变量的面之值，然后把值存放于内存。

F_PROFILE 是随 DEFINE_PROFILE 一同应用的一个比较典型的宏，并且它是由 FLUENT 所提供的预先定义的宏。对于一个给定的面和脉络，F_PROFILE 把一边界条件保存在内存中，它则被套入了面的轮循里。F_PROFILE 里的指标 i 与 DEFINE_PROFILE 的是同一个。F_PROFILE 使用脉络指示器 t，面的标识符 f 和指标 i 在内存里设置合适的边界面。

在多相的情况下，一个 DEFINE_PROFILE 可能不止被调用一次。为避免其发生，可在 UDF 的名字中添加预固定 MP_，那么这个函数将仅被访问一次，即使它被用于多剖面情形。

- 例子

下面名为 pressure_profile 的 UDF，根据下面的方程生成了一个比喻的压力剖面，方程为：

$$p(y) = 1.1\times10^5 - 0.1\times10^5\left(\frac{y}{0.0745}\right)^2$$

注意，这个 UDF 假设以边界区域的几何中心为原点生成网格，其中这个边界区域正是 UDF 所作用的区域。y 在进口的中心上取 0.0，从进口的顶部到底部其取值范围是 ±0.0745。源代码可以在 FLUENT 中被通译或编译。

```
/**********************************************************************
对于一个涡轮叶片，规定稳态的比喻压力剖面为其边界剖面的 UDF
**********************************************************************/
    #include "udf.h"

    DEFINE_PROFILE(pressure_profile, t, i)
    {
      real x[ND_ND];
      real y;
      face_t f;

      begin_f_loop(f, t)
        {
          F_CENTROID(x,f,t);
          y = x[1];
          F_PROFILE(f, t, i) = 1.1e5 – y*y/(.0745*.0.745)*0.1e5;
        }
      end_f_loop(f, t)
    }
```

这个名为 pressure_profile 的函数有两个自变量：t 和 i。t 是面的脉络的一个指示器，是一个整数，它是在每一轮循中被设置的变量的一个数值标签。

在函数体里，变量 f 被宣布为一个面。一个一维数组 x 和变量 y 的数据类型被宣布为实型。在宣布了变量之后，一个轮循的宏被用来轮循过区域的每一个面去创建一个剖面或一组数据。在每一轮循中，F_CENTROID 的返回值是指标为 f 的面的质心（数组 x）。保存在 x[1]

里的坐标 y 被指定为变量 y，然后它被用来计算压力。接着，这个值被用于 F_PROFILE，F_PROFILE 使用整数 I 在内存中给出压力面的值。

13. DEFINE_PROPERTY

可以使用 DEFINE_PROPERTY 宏为单相流和多相流定义一个自定义材料特性。当为一个混合材料写一个用户定义混合法则的时候，用户需要使用特殊工具获得物种的材料特性。如果想定义一个用户质量扩散特性，必须使用 DEFINE_DIFFUSIVITY 而不是 DEFINE_PROPERTY，详细细节在本节的前面有介绍。

使用 DEFINE_PROPERTY 宏，可以用户化的一些特性为：

- 密度（仅作为一个温度的函数）
- 粘性系数
- 热传导率
- 吸收和发散系数
- 层流流动速度
- 应变率
- 粒子或小滴直径（多相混合物模型）
- 用户为密度，粘性系数和混合物材料的传导系数定义的混合法则
- 表面张力系数（多相 VOF 模型）

值得注意的是，UDFs 不能用来定义比热特性；在 FLUENT 中，比热数据只可以被调用而不允许修改。

- 应用

宏：	DEFINE_PROPERTY (name, c, t)
自变量类型：	cell_t c
	Thread *t
函数返回：	real

DEFINE_PROPERTY 中有三个自变量：name，c 和 t。name 是用户给出的 UDF 的名字。c 和 t 是被 FLUENT 的求解器传送到 UDF 的变量。UDF，仅对于一个单独的单元，将需要计算它的实型特性，然后把这个值返回到求解器。被传送的变量，t，是脉络的一个指示器，其中边界条件在这个脉络上应用。c 是一个指数，它被用来鉴别特性函数所使用的单元。

注意，与源项 UDFs 相同，特性 UDFs 被 FLUENT 从单元脉络上的一个轮循内访问。自从特性被分配到一个单元上，求解器传送所有所需要的变量让一个 DEFINE_PROPERTY UDF 去定义一个用户材料。从而，自从 FLUENT 已经工作起，UDF 将不需要轮循过一个区域里的单元。

- 辅助工具

在下面介绍了关于 UDFs 特性的通常使用的一些辅助工具，它们是 generic_property，MATERIAL_PROPERTY，THREAD_MATERIAL 和 mixture_species_loop。

函数：	generic_property (name, c, t, prop, id, T)
自变量类型：	cell_t c
	Thread *t
	Property *prop
	Property_ID id
	real T
函数返回：	real

generic_property 是一个总目标函数，关于所给定的脉络材料的给定特性 id，这个

函数的返回值是实型的。

t 是单元脉络的一个指示器，c 是一个指数，它被用来鉴别特性函数所使用的单元。prop 是关于脉络材料的特性排列的一个指示器，它可以通过宏 MATERIAL_PROPERTY(m)获得。T 是求特性之值所用的温度，它仅在规定了一个多项式方法的情况下使用。也许你想对所要求的特性定义一个用户混合法则，而 id 是这个特性的 Property_ID（例如，关于热传导系数的 PROP_ktc）。所使用的 Property_ID 有：

- PROP_rho，密度
- PROP_mu，粘性系数
- PROP_ktc，热传导系数

宏：　　MATERIAL_PROPERTY (m)

自变量类型：　　Material *m

MATERIAL_PROPERTY 定义于 materials.h 中，并返回到关于给定的材料指示器 m 的特性排列 prop 的一个指示器。

宏：　　THREAD_MATERIAL (t)

自变量类型：　　Thread *t

THREAD_MATERIAL 定义于 threads.h 中，并返回到与所给定的单元脉络 t 相联系的 Material 的指示器 m。

注意，在 FLUENT 以前的版本里，THREAD_MATERIAL 有两个自变量（t，i），而现在仅有一个变量（t）。

宏：　　mixture_species_loop (m，sp，i)

自变量类型：　　Material *m，Material *sp，int I

mixture_species_loop 定义于 materials.h，并轮循过所有的关于给定混合材料的物种。

- 例子—依赖于温度的粘性系数之特性

下面名为 cell_viscosity 的 UDF，生成了一些模拟凝固的粘性系数剖面。关于区域中的每一个单元，函数都被访问。暖流（$T > 288K$）中的粘性系数有一个关于液体（$5.5\times10^{-3} kg/(m\cdot s)$）的分子值，而关于冷却器区域（$T < 286K$）的粘性系数则有一个大的多的数值（$1.0kg/(m\cdot s)$）。在媒介温度范围内（$286K \le T \le 288K$），粘性系数的范围在上述两个值之间呈线性变化：

$$\mu = 143.2135 - 0.49725T \qquad (18.2.3.14)$$

这个模型是在假设随着液体的冷却与其粘性的增大，它的速度将减小，所以模型模拟凝固。这里，没有对能量场所包含的冷冻潜在热量做修正。源代码可以在 FLUENT 中被通译或编译。

```
/***********************************************************************
UDF：通过规定一个依赖于温度的粘性特性来模拟凝固
***********************************************************************/
    DEFINE_PROPERTY(cell_viscosity, c, t)
    {
      real mu_lam;
      real temp = C_T(c, t);

      if (temp > 288.)
        mu_lam = 5.5e-3;
```

```
  else if (temp > 286.)
    mu_lam = 143.2135 – 0.49725 * temp;
  else
    mu_lam = 1.;

  return mu_lam;
}
```

这个名为 cell_viscosity 的函数被定义在一个单元上。引入了两个实型变量：C_T(c，t)的值，temp，和方程所计算的层流粘性系数，mu_lam。检查温度的值，并且基于 mu_lam 的值下降的范围，计算合适的 mu_lam 之值。在函数的最后，关于粘性系数的计算值被返回到求解器。

14. DEFINE_SCAT_PHASE_FUNC

可以使用 DEFINE_SCAT_PHASE_FUNC 宏为离散纵坐标（DO）模型定义辐射分散相函数。这个函数计算两个值：辐射能量从 i 方向到 j 方向的分散量所占的组份，前进分散因子。

- 应用

宏：	DEFINE_SCAT_PHASE_FUNC (name, cosine, f)
自变量类型：	real cosine
	real *f
函数返回：	real

DEFINE_SCAT_PHASE_FUNC 中有三个自变量：name，cosine 和 f。name 是用户给出的 UDF 的名字。cosine 和 f 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要计算辐射能量从 i 方向到 j 方向的分散量所占的组份，其中计算中数值的类型是实型的，UDF 把这个实型值返回到求解器。

被传送的变量 cosine 是一个实型变量，它是两个方向的夹角余弦。f 是一个实型指示器，它指向内存中存储实型前进分散因子的位置。注意，通过访问每一对唯一的离散纵坐标的函数，求解器计算和保存每一种材料的一个分散矩阵。

- 例子

下面的这个例子，在一个单独的 C 语言源代码文件里含有几个 UDFs。源代码可以在 FLUENT 中被通译或编译。

```
/***********************************************************************
UDFs：执行前进和后退的分散相函数，其中这个函数被 Jendoubi  等引用
***********************************************************************/
    #include “udf.h”

    DEFINE_SCAT_PHASE_FUNC(ScatPhiB2, c, fsf)
    {
      real phi=0;
      *fsf = 0;
      phi = 1.0-1.2*c+0.25*(3*c*c-1);
      return (phi);
    }
    DEFINE_SCAT_PHASE_FUNC(ScatPhiB1, c, fsf)
    {
      real phi=0;
```

```
  *fsf = 0;
  phi = 1.0 – 0.56524*c + 0.29783*0.5*(3*c*c-1)+
    0.08571*0.5*(5*c*c*c-3*c) + 0.01003/8*(35*c*c*c*c-30*c*c+3) +
    0.00063/8*(63*c*c*c*c*c-70*c*c*c+15*c);
  return (phi);
}
DEFINE_SCAT_PHASE_FUNC(ScatPhiF3, c, fsf)
{
  real phi=0;
  *fsf = 0;
  phi = 1.0 + 1.2*c + 0.25*(3*c*c-1);
  return (phi);
}
DEFINE_SCAT_PHASE_FUNC(ScatPhiF2, c, fsf)
{
  real phi=0;
  real coeffs[9]={1,2.00917,1.56339,0.67407,0.22215,0.04725,
                 0.00671,0.00068,0.00005};
  real P[9];
  int i;
  *fsf = 0;
  P[0] = 1;
  P[1] = c;
  phi = P[0]*coeffs[0] + P[1]*coeffs[1];
  for(i=1;i<7;i++)
    {
      P[i+1] = 1/(i+1.0)*((2*i+1)*c*P[i] – i*P[i-1]);
      Phi += coeffs[i+1]*P[i+1];
    }
  return (phi);
}
DEFINE_SCAT_PHASE_FUNC(ScatIso, c, fsf)
{
  *fsf=0;
  return (1.0);
}
```

15. DEFINE_SOURCE

可以使用 DEFINE_SOURCE 宏为 FLUENT 中不同类型被求解的输运方程规定用户源项，包括：

- 连续
- 动量
- k， ε
- 动能（也可以对于多边形区域）
- 物种质量组份
- 用户定义标量（UDS）输运

- 应用

宏： DEFINE_SOURCE (name, c, t, dS, eqn)

自变量类型：　　cell_t c

Thread *t

real dS[]

int eqn

函数返回：　　real

DEFINE_SOURCE 中有三个自变量：name，c，t，dS 和 eqn。name 是用户给出的 UDF 的名字。c，t，dS 和 eqn 是被 FLUENT 的求解器传送到 UDF 的变量。

被传送的变量 t 是单元脉络的一个指示器，c 是一个指数，它规定源项所使用单元。排列 dS 对输运方程中的独立变量规定了源项的派生类。如果这些源项提高了稳定性，那么它们可能被用来线性化源项。为了使大家更清楚其意思，（18.2.3.14）式给出了源项的一般表达式，这里ϕ是独立变量，A是源项的显示部分，$B\phi$是隐示部分

$$S_{\phi} = A + B\phi \qquad (18.2.3.15)$$

由于增加了解矩阵的对角线上的项，方程（18.2.3.15）中规定关于 B 的一个值能够提高解的稳定性和收敛率。FLUENT 自动地确定用户所给 B 的值是否将有助于解的稳定。如果能够，那么 FLUENT 将定义 A 为$S^* - (\partial S/\partial\phi)^* \phi^*$，B 为$(\partial S/\partial\phi)^*$。如果不能，源项被明确地处理。

UDF 将需要仅就一个单元计算实型源项，并返回值于求解器，但你做了关于设置隐示项 dS 为$dS/d\phi$的选择，或者通过设置它为 0.0 而施加于源项的显示解。

注意，和特性 UDFs 类似，源项 UDFs（使用 DEFINE_SOURCE 定义的）被 FLUENT 从单元脉络上的一个轮循里访问。由于源项在一个单元为基地被求解，求解器把需要定义的一用户源项的所有变量传送到 DEFINE_SOURCE 项的 UDF。从而，UDF 将不需要轮循过脉络中的单元，事实上，FLUENT 已做了此项工作。

所有源项的单位使用的形式是$生成率/体积$。例如，联系方程的一个源项有单位$kg/(m^3 \cdot s)$。

- 例子

下面这个名为 xmom_source 的 UDF 例子，在 FLUENT 中用来添加源项。源代码可以在 FLUENT 中被通译或编译。函数产生一个 x-动量源项，它随 y 位置的变化为

$$source = -0.5 C_2 \rho y |v_x| v_x$$

假设

$$source = S = -A |v_x| v_x$$

其中

$$A = 0.5 C_2 \rho y$$

那么

$$\frac{dS}{dv_x} = -A|v_x| - A v_x \frac{d}{dv_x}(|v_x|)$$

被返回的源项是

$$source = -A\,|\,v_x\,|\,v_x$$

源项关于 v_x 的导数是

$$\frac{dS}{dv_x} = -2A\,|\,v_x\,|$$

```
/***************************************************************************
UDFs ： 在 一 个 空 间 多 孔 介 质 中 规 定 一 个 x- 动 量 源 项
***************************************************************************/
#include “udf.h”

#define C2 100.0

DEFINE_SOURCE(xmom_source, c, t, dS, ean)
{
  real x[ND_ND];
  real con, source;

  C_CENTROID(x, c, t);
  con = C2*0.5*C_R(c, t)*x[1];

  source = -con*fabs(C_U(c, t))*C_U(c, t);
  dS[eqn] = -2.*con*fabs(C_U(c,t));

  return source;
}
```

16. DEFINE_SR_RATE

可以使用 DEFINE_SR_RATE 宏定义一个用户表面反应率。使用这个宏定义的用户表面反应率将覆盖面板 **Materials** 中的默认反应率（例如，finite-rate）。下面将介绍一个关于取决于气体物种质量组份的一个反应率的例子。同时也介绍了一个考虑地位物种的反应率 UDF。

- 应用

宏：	DEFINE_SR_RATE (name, f, t, my, yi, rr)
自变量类型：	face_t f
	Thread *t
	Reaction *r
	real *mw
	real *yi
	real *rr
函数返回：	void

DEFINE_SR_RATE 中有七个自变量：name，f，t，r，my，yi 和 rr。name 是用户给出的 UDF 的名字。一旦 UDF 被编译或连接之后，用户为函数所选的名字将在 FLUENT 中图

形用户界面变地可视与可选。f，t，r，my，yi 和 rr 是被 FLUENT 的求解器传送到 UDF 的变量。

被传送的变量 t 是表面反应率所使用的单元脉络的一个指示器，f 是一个指数，它被用来鉴别所给脉络里的一个面或者一个多孔区域里表面反应情况下的单元。r 是关于反应的数据结构的一个指示器。mw 是包含物种分子重量的一个实型排列的一个指示器。yi 也是一个实型排列的指示器，其中这个排列包含在表面上的气体物种质量组份和地位物种的质量组份。

UDF 将需要通过实型指示器 rr 设置参考值的反应率。

- 例子

下面这个名为 arrhenius 的通译的 UDF，在 FLUENT 中使用物种质量组份定义一个用户表面反应率。

```
/***********************************************************************
用户表面反应率的 UDF
***********************************************************************/
#include “udf.h”

/*ARRHENIUS CONSTANTS*/
#define PRE_EXP 1e+15
#define ACTIVE 1e+08
#define BETA 0.0

real arrhenius_rate(real temp)
{
  return
PRE_EXP*pow(temp,BETA)*exp(-ACTIVE/)UNIVERSAL_GAS_CONSTANT*temp));
}

/*物种数目。在 FLUENT 面板中顺序必须匹配*/
#define HF  0
#define WF6 1
#define H20  2
#define NUM_SPECS 3

/*Reaction Rate Routine 被用在两个 UDFs 中*/
real reaction_rate(cell_t c,Thread *cthread,real mw[],real yi[])
{
  real concenHF = C_R(c, cthread)*yi[HF]/mw[HF];
  return arrhenius_rate(C_T(c,cthread))*pow(concenHF,HF_EXP);
}

DEFINE_SR_RATE(arrhenius,f,fthread,r,mw,yi,rr)
{
  *rr = reaction_rate(F_CO(f,fthread),THREAD_TO(f,fthread),mw,yi);
}

real contact_area(cell_t c,Thread *t,int s_id, int *n)
{
  int I = 0;
```

```
    real area =0.0, A[ND_ND];

    *n = 0;
    c_face_loop(c,t,I)
    {
      if(THREAD_ID(C_FACE_THREAD(c,t,I)) == s_id)
         {
             (*n)++;
             F_AREA(A, C_FACE(c,t,I), C_FACE_THREAD(c,t,I));
             area += NV_MAG(A);
         }
     }
     return area;
  }
```

17. DEFINE_TURB_PREMIX_SOURCE

使用 DEFINE_TURB_PREMIX_SOURCE 宏可以在预混合物燃烧模型和部分预混合物燃烧模型中自定义一个用户湍流燃烧速度。

- 应用

宏：	DEFINE_TURB_PREMIX_SOURCE (name, c, t, turb_flame_speed, source)
自变量类型：	face_t f
	Thread *t
	real *turb_flame_speed
	real *source
函数返回：	void

DEFINE_TURB_PREMIX_SOURCE 中有五个自变量：name，c，t，turb_flame_speed 和 source。name 是用户给出的 UDF 的名字。c，t，turb_flame_speed 和 source 是被 FLUENT 的求解器传送到你的 UDF 的变量。你的 UDF 将需要设置湍流燃烧速度为 turb_flame_speed 指示器所参考的值。它还将需要设置源项为源指示器所参考的值。

被传送的变量 t 是脉络的一个指示器，其中湍流预混合物源项在这个脉络上被应用。f 是一个指数，它被用来鉴别所给脉络里的一个单元。turb_flame_speed 是湍流燃烧速度的一个实型指示器，source 是反应发展源项的一个实型指示器。

- 例子

下面这个名为 turb_flame_src 的 UDF，在预混合物燃烧模型中规定了一个用户湍流速度和源项。源代码必须作为一个编译的 UDF 在 FLUENT 中被执行。

在 FLUENT 的标准的预混合物燃烧模型中，过程变量的平均反应率的数学模型为

$$\rho S_c = \rho_u U_t \,|\nabla_c| \qquad (18.2.3.16)$$

这里，c 是平均反应过程变量，ρ 是密度，U_t 是湍流燃烧速度。

在这个 UDF 的例子中，湍流燃烧速度定义为

$$U_t = U_l \sqrt{1 + (u'/U_l)^2} \qquad （18.2.3.17）$$

这里，U_t 是层流燃烧速度，u' 是湍流波动。

```
/*********************************************************************
      为预混和物燃烧模型规定一个用户湍流燃烧速度和源的 UDF
*********************************************************************/
 #include "udf.h"
 #include "sg_pdf.h"/*不包含在 udf.h 中，所有必须包含在这个文件中*/

 DEFINE_TURB_PREMIX_SOURCE(turb_flame_src,c,t,turb_flame_speed,
   source)
 {
   real up = TRB_VEL_SCAL(c,t);
   real ut,ul,grad_c,rho_u,DV[ND_ND];

   ul = UL_par_premix(C_FMEAN(c,t));
   rho_u = rho_par_premix(C_FMEAN(c,t));

   if(NNULLPP(THREAD_STORAGE(t,SV_PREMIXC_G)))
    {
      NV_V (DV, = , C_STORAGE_R_NV(c,t,SV_PREMIXC_G));
      grad_c = sqrt(NV_DOT(DV,DV));
    }
   ut = ul*sqrt( 1.+ SQR(up/ul));
   *turb_flame_speed = ut;
   source = rho_u*ut*grad_c;
 }
```

18. DEFINE_TURBLENT_VISCOSITY

可以使用 DEFINE_TURBLENT_VISCOSITY 宏为 Spalart-Allmaras，$k-\varepsilon$，$k-\omega$ 和 LES 湍流模型定义一个用户湍流粘性系数。

- 应用

宏：	DEFINE_TURBLENT_VISCOSITY (name, c, t)
自变量类型：	cell_t c
	Thread *t
函数返回：	real

DEFINE_TURBLENT_VISCOSITY 中有三个自变量：name，c 和 t。name 是用户给出的 UDF 的名字。c 和 t 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要把湍流粘性系数的实型值返回到求解器。

被传送的变量 t 是脉络的一个指示器，其中湍流粘性系数应用于这个脉络上。c 是一个指数，它被用来鉴别所给脉络里的一个单元。

- 例子

下面这个名为 user_mu_t 的 UDF，定义关于标准 $k-\varepsilon$ 湍流模型的一个用户湍流粘性系数。注意，例子中的 M_keCmu 的值是通过图形用户界面来定义的，但所有的 UDFs 都可以访问它。源代码在 FLUENT 中可以被通译或编译。

```
/*********************************************************************
UDF: 使用 DEFINE_TURBULENT_VISCOSITY 为标准 k-ε 湍流模型规定一个用
```

户湍流粘性系数
***/

```
#include “udf.h”

DEFINE_TURBULENT_VISCOSITY(user_mu_t, c, t)
{
  real mu_t;
  real rho = C_R(c, t);
  real k = C_K(c, t);
  real d = C_D(c, t);

  mu_t = M_keCmu*rho*SQR(k)/d;

  return mu_t;
}
```

19. DEFINE_UDS_FLUX

当打算用户化自定义标量（UDS）输运方程中的平流通量项的计算方法时，可以使用 DEFINE_UDS_FLUX 宏。

- 应用

宏：	DEFINE_UDS_FLUX (name, f, t, i)
自变量类型：	face_t f
	Thread *t
	int i
函数返回：	real

DEFINE_UDS_FLUX 中有四个自变量：name，f，t，和 i。name 是用户给出的 UDF 的名字。f，t，和 i 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要把质量流量率的实型值通过所给的面返回到求解器。

被传送的变量 t 是脉络的一个指示器，其中用户定义标量通量应用于这个脉络上。f 是一个指数，它被用来鉴别所给脉络里的一个面。i 是一个用来鉴别用户定义标量的指数。

微分输运方程中的平流项有最一般的形式：

$$\nabla \cdot \vec{\psi}\phi \qquad (18.2.3.18)$$

这里ϕ是用户定义标量守恒量，$\vec{\psi}$是一个矢量场。在默认的平流项中，$\vec{\psi}$被默认为标量密度和矢量速度的成绩：

$$\vec{\psi}_{default} = \rho\vec{v} \qquad (18.2.3.19)$$

为了使用 DEFINE_UDS_FLUX 定义方程（18.2.3.18）中的平流项，UDF 需要把标量值$\vec{\psi} \cdot \vec{A}$到 FLUENT，其中$\vec{A}$是面的法线矢量。

注意，由 UDF 提供的平流通量场应当为自由散度。在不连续的项中，过每一单元的所有面的总通量的平均值应当为零。如果平流场不是自由散度，那么ϕ不被保存，并且将导致

单元值ϕ内出现 overshoots/undershoots 的情况。

例如，需要计算 UDF 中的$\vec{\psi}$，其中使用由 FLUENT 提供的关于速度和密度的预定义宏或者使用自己指定的命令。在下面的例子中，使用的是前一种方法。

注意，若所求解的不止一个标量，则可以在 UDF 中使用一个 if 声明的条件去定义关于每个 i 的不同的通量函数。i=0 与标量 0 相联系。还应当注意的是，$\vec{\psi}\cdot\vec{A}$的单位必须是 SI 中的质量流（例如，kg/s）。

- 例子

下面这个名为 my_uds_flux 的 UDF，通过一个给定的面返回值是质量流率。通常，这个通量通过由 FLUENT 提供的宏 F_FLUX（f, t）就可以得到。如果流动方向与面的法线方向一致，那么由 FLUENT 的求解器计算得到的通量之符号是正的，当然，流动方向与面的法线方向相反，则是负的。按惯例，面的法线通常是指向边界面的区域的，在面的内部其方向是从 c0 到 c1 的。源代码必须作为一个编译的 UDF 被执行。

```
/*************************************************************************
UDF：使用 DEFINE_UDS_FLUX 执行标量输运方程中的一个简化的平流项
*************************************************************************/
 #include “udf.h”

 DEFINE_UDS_FLUX(my_uds_flux, t, i)
 {
   Thread *t0, *t1 = NULL;
   cell_t c0, c1 = -1;

   real NV_VEC(psi_vec), NV_VEC(A);

   /*面 f 的相邻单元，和它们的脉络*/

   t0 = THREAD_TO(f,t);
   c0 = F_CO(f,t);

   if(NULL!= THREAD_T1(f,t))
   /*二选一：如果(! BOUNDARY_FACE_THREAD_P(t))*/
    {
      t1 = THREAD_T1(f,t);
      c1 = F_C1(f,t);
    }
   else
    {
      t1 = NULL;
      c1 = -1;
    }

   /*如果面位于域的边界上，使用面的值；*/
   /*如果面位于 IN 域，使用邻近单元的平均值。*/
```

```
  if (NULL == t1)
  /*二选一：如果 (BOUNDARY_FACE_THREAD_P(t))*/
  {
    NV_D(psi_vec, =, F_U(f,t),F_V(f,t),F_W(f,t));
    NV_S(psi_vec, *= F_R(f,t));
   }
  else
   {
    NV_D(psi_vec, =, C_U(c0,t0),C_V(c0,t0),C_W(c0,t0));
    NV_D(psi_vec, +=, C_U(c1,t1),C_V(c1,t1),C_W(c1,t1));
    NV_S(psi_vec, /=, 2.);/*平均*/;
    NV_S(psi_vec, *=, (((C_R(c0,t0) + C_R(c1,t1)/2.)));
   }

  /*现在 psi_vec 包含上面的"psi"*/
  /*下一步，取得面的法线矢量*/

  F_AREA(A,f,t);

  /*最后，返回两个的点乘*/
  /*Fluent 将乘以这个返回值*/
  /*通过 phi_f(面上的标量值)*/
  /*获得完全的平流项…*/

  return NV_DOT(psi_vec, A);
}
```

20. DEFINE_UDS_UNSTEADY

当打算用户化自定义标量（UDS）输运方程中的非稳态项时，可以使用 DEFINE_UDS_UNSTEADY 宏。

- 应用

宏：	DEFINE_UDS_UNSTEADY (name, c, t, i, apu, su)
自变量类型：	cell_t c
	Thread *t
	int I
	real *apu
	real *su
函数返回：	void

DEFINE_UDS_UNSTEADY 中有六个自变量：name，c，t，i，apu 和 su。name 是用户给出的 UDF 的名字。c，t，i，apu 和 su 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要分别设置被实型指示器 apu 和 su 所引用的非稳态项的值到中心系数和源项。

被传送的变量 t 是脉络的一个指示器，其中 UDS 输运方程的非稳态项应用于这个脉络上。c 是一个指数，它被用来鉴别所给脉络里的一个单元。i 是一个用来鉴别用户定义标量的指数，其中这个标量是由非稳态项设置的。

FLUETN 求解器希望瞬时项会被分解成一个源项，su，和一个中心系数，apu。方程中的这些项的设置方式与一个源项的隐示和显示成分可能被处理的方式是类似的。所以，把非

稳态项移到右边并离散如下：

$$
\begin{aligned}
unsteady\ term &= -\int \frac{\partial}{\partial t}(\rho\phi)dV \\
&\approx -\left[\frac{\rho\phi)^n - (\rho\phi)^{n-1}}{}\right]\cdot \Delta V \\
&= \underbrace{-\frac{\rho\Delta V}{\Delta t}\phi^n}_{apu} + \underbrace{\frac{\rho\Delta V}{\Delta t}\phi^{n-1}}_{su}
\end{aligned}
\qquad (18.2.3.20)
$$

方程（18.2.3.20）给出了 su 和 apu 的定义方式。注意，如果所求解的不止一个标量，那么可以在 UDF 中使用一个 if 声明的条件去定义关于每个 i 的一个不同的非稳态项。i=0 与标量 0 相联系。

● 例子

下面这个名为 uns_time 的 UDF，使用 DEFINE_UDS_UNSTEADY 修改用户定义的时间标量的导数。源代码在 FLUENT 中可以被通译或编译。

```
/*************************************************************************
UDF：规定用户定义的时间标量的导数
*************************************************************************/
#include "udf.h"

DEFINE_UDS_UNSTEADY(uns_time, c, t, i, apu, su)
{
  real physical_dt,vol,rho,phi_old;
  physical_dt = RP_Get_Real("physical-time-step");
  vol =C_VOLUME(c,t);

  rho = C_R_M1(c,t);
  *apu = -rho*vol/physical_dt;/*隐示部分*/
  phi_old = C_STORAGE_R(c,t,SV_UDSI_M1(i));
  *su = rho*vol*phi_old/physical_dt;/*显示部分*/
}
```

21. DEFINE_VR_RATE

若想为一个单独反应或多个反应定义一个用户测定体积的反应，则可以使用 DEFINE_VR_RATE 宏。在 FLUENT 执行过程中，对于每个单元中的每一反应 DEFINE_VR_RATE 都被调用。

● 应用

宏：	DEFINE_VR_RATE (name, c, t, r, mw, yi, rr, rr_t)
自变量类型：	cell_t c
	Thread *t
	Reaction *r
	real *mw
	real *yi
	real *rr
	real *rr_t
函数返回：	void

DEFINE_VR_RATE 中有八个自变量：name，c，t，r，mw，yi，rr 和 rr_t。name 是用户给出的 UDF 的名字。c，t，r，mw，yi，rr 和 rr_t 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要分别设置被实型指示器 rr 和 rr_t 所引用的值到层流和湍流反应率。

被传送的变量 t 是脉络的一个指示器，其中测定体积的反应率应用于这个脉络上。c 是一个指数，它被用来鉴别所给脉络里的一个单元。r 是一个到代表当前反应的数据结构的指示器。mw 是一个物种分子重量的实型指示器排列，yi 是一个物种质量组份的指示器排列。

rr 和 rr_t 被计算，并且较小的一个值在有限率或漩涡耗散化学反应机制被使用的时候被应用。注意，rr 和 rr_t 是单位为 $kg \cdot mol/(m^3 \cdot s)$ 的转化率。当它们分别乘上化学当量计算系数时，它们产生各自的化学组成部分的生成率或消耗率。

- 例子

下面这个名为 vol_reac_rate 的 UDF，指定了一个体积反应率。源代码在 FLUENT 中必须作为编译的 UDF 被执行。

```
/**********************************************************************
规定一个体积反应率的 UDF
关于反应率，Fluent 计算的要素为：计算的仅是一个 Arrhenius(“有限率“)反应率，
它是从图形用户界面中用户所给的输入值开始计算的
**********************************************************************/
 #include “udf.h”

 DEFINE_VR_RATE(vol_reac_rate, c, t, r, wk, yk, rate, rr_t)
 {
   real ci,prod;
   int i;

 /*计算 Arrhenius 反应率*/

   prod = 1.;

   for(i = 0;i<r->n_reactants;i++)
    {
      ci = C_R(c,t) * yk[r->reactant[i]]/wk[r->reactant[i]];
      prod *= pow(ci, r->exp_reactant[i]);
    }
   *rate = r->A * exp(-r->E/(UNEVERSAL_GAS_CONSTANT *
                 C_T(c,t)))*pow(C_T(c,t),r->b) * prod;
   *rr_rate = *rate;
   /*没有返回值*/
   vol =C_VOLUME(c,t);
 }

   rho = C_R_M1(c,t);
   *apu = -rho*vol/physical_dt;/*隐示部分*/
   phi_old = C_STORAGE_R(c,t,SV_UDSI_M1(i));
   *su = rho*vol*phi_old/physical_dt;/*显示部分*/
 }
```

18.3.4: 多相 DEFINE 宏

多相 DEFINE 宏仅用于多相应用。表 18.2.4.1 给出了一个关于 DEFINE 宏、DEFINE 宏定义的函数、以及 FLUENT 中被激活的面板的快捷参考指导。每一个 DEFINE 宏的定义都被放在了 udf.h 头文件中。为方便读者，把它们列在了附录 A 中。

此模型主要包括以下几种特殊的 DEFINE 宏：

- DEFINE_CAVITATION_RATE
- DEFINE_EXCHANGE_PROPERTY
- DEFINE_VECTOR_EXCHANGE_PROPERTY

表 18.2.4.1：关于物种模型的 DEFINE 宏的快捷参考指导

模型	函数	DEFINE 宏	激活的面板
VOF	质量输运	DEFINE_EXCHANGE_PROPERTY	Phase Interaction
Mixture	质量输运	DEFINE_EXCHANGE_PROPERTY	Phase Interaction
	阻力（拖拉）系数	DEFINE_EXCHANGE_PROPERTY	Phase Interaction
	滑动速度	DEFINE_VECTOR_EXCHANGE_PROPERTY	Phase Interaction
	空化率	DEFINE_CAVITATION_RATE	User-Defined Function Hooks
Eulerian	质量输运	DEFINE_EXCHANGE_PROPERTY	Phase Interaction
	热传递	DEFINE_EXCHANGE_PROPERTY	Phase Interaction
	阻力系数	DEFINE_EXCHANGE_PROPERTY	Phase Interaction
	升力系数	DEFINE_EXCHANGE_PROPERTY	Phase Interaction

1. DEFINE_CAVITATION_RATE

可以使用 DEFINE_CAVITATION_RATE 宏对一个多相混合模型流动建立由于压力张力而导致的产生水蒸气的模型。

- 应用

宏：　DEFINE_CAVITATION_RATE (name, c, t, p, rhoV, rhoL, vofV, p_v, n_b, m_dot)

自变量类型：
cell_t c
Thread *t
real *p
real *rhoV
real *rhoL
real *vofV
real *p_v
real *n_b
real *m_dot

函数返回：　void

DEFINE_CAVITATION_RATE 中有十个自变量：name，c，t，p，rhoV，rhoL，vofV，

p_v，n_b 和 m_dot。name 是用户给出的 UDF 的名字。c，t，p，rhoV，rhoL，vofV，p_v，n_b 和 m_dot 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要设置气蚀率为实型指示器 m_dot 所参考的值。

被传送的变量 t 是一个均衡混合物脉络的指示器。c 是一个指数，它被用来鉴别给定脉络内的一个单元。其余的变量都是以下数据的实型指示器：共享压力（p），水蒸气密度（rhoV），液体密度（rhoL），水蒸气体积组份（vofV），汽化器压力（p_v），单位体积的气泡数（n_b），和水蒸气形成率（m_dot）。

- 例子

下面名为 user_cavitation._rate 的 UDF，计算关于一个多相混合物中的液体和水蒸气相（气蚀率）之间的质量转换。

```
/***********************************************************************
计算气蚀率的 UDF
***********************************************************************/
 #include “udf.h”

 #define MIN_VOF 1.e-5
 #define MAX_VOF 0.999999

 DEFINE_CAVITATION_RATE(user_cavitation_rate,c,t,p,rhoV,
                        rhoL,vofV,p_v,n_b,m_dot)
 {
   real p_vapor = *p_v;
   real n_bubbles = *n_b;
   real dp, vofM, radV;
   dp =p_vapor – ABS_P( p[c], op_pres );
   vofM = MIX(MAX(MIN_VOF,vofV[c],MAX_VOF);
   radV = pow(vofM/((1.-vofM)*4./3.*M_PI*n_bubbles),1./3.);
   if(dp>0.)
       *m_dot = (1.-vofV[c])*n_bubbles*4.*M_PI*radV*
                    radV/(1.*n_bubbles*4./3.*M_PI*radV*radV*radV)*
                    sqrt(2.*ABS(dp)/(3.*rhoL[c]));
   else
       {
         *m_dot = -(1.-vofV[c])*n_bubbles*4.*M_PI*radV
                    radV/(1.*n_bubbles*4./3.*M_PI*radV*radV*radV)*
                    sqrt(2.*ABS(dp)/(3.*rhoL[c]));
         if(vofV[c]<=MIN_VOF)*m_dot=0.;
       }
 }
```

2. DEFINE_EXCHANGE_PROPERTY

使用 DEFINE_EXCHANGE_PROPERTY 宏可以规定关于多相模型中的相交互变量的 UDFs。这些变量包括热和质量的转换函数，阻力和升力系数。关于 FLUENT 中的多相模型，可以使用 DEFINE_EXCHANGE_PROPERTY 宏规定的用户定义函数列在了下面的表（18.2.4.2）中。注意，相交互变量的表面张力系数要用宏 DEFINE_PROPERTY 来定义的。

表 18.2.4.2：DEFINE_EXCHANGE_PROPERTY 变量

VOF 模型	混合物模型	欧拉模型
质量转换	质量转换 阻力系数	质量转换 热转换 阻力系数 升力系数

- 应用

宏： DEFINE_EXCHANGE_PROPERTY (name, c, mixture_thread, second_column_phase_index, first_column_phase_index)

自变量类型： cell_t c
Thread *mixture_thread
int second_column_phase_index
int first_column_phase_index

函数返回： real

DEFINE_EXCHANGE_PROPERTY 中有五个自变量：name，c，mixture_thread，second_column_phase_index 和 first_column_phase_index。name 是用户给出的 UDF 的名字。其它四个变量是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要把升力系数，阻力系数，热或质量转换的值以实型的形式返回到求解器。

mixture_thread 是均衡混合物的脉络的一个指示器。c 是脉络上一个单元的指数，其中 mixture_thread 指向这个脉络。second_column_phase_index 和 first_column_phase_index 是相应于多相流中你所指定的一个滑动速度的双相的整数标识符。标识符与在图形用户界面中的 **Phase Interaction** 面板中所选择的相是一致的。一个 0 指数与基本相一致，每两相增加 1。

- 例子

下面名为 custom_drag 的 UDF，可以用来把 FLUENT 中的缺省 Syamlal 阻力法则用户化。对于 bfac，缺省的阻力法则使用的是 0.8（对于 void<=0.85）和 2.65（void>0.85）。使用关于 bfac 的参数 0.28 和 9.07 都会导致一个最小的流动速度 8cm/s。

```
/*********************************************************************
UDF：在 Fluent 中，用户化缺省的 Syamlal 阻力法则
*********************************************************************/
 #include “udf.h”

 #define pi 4.*atan(1.)
 #define diam2 3.e-4

 DEFINE_EXCHANGE_PROPERTY(custom_drag, cell, mix_thread, s_col, f_col)
 {
    Thread *thread_g, *thread_s;
    real x_vel_g, x_vel_s, y_vel_g, y_vel_s, abs_v,slip_x, slip_y,
       rho_g, rho_s, mu_g,reyp, afac,
       bfac, void, vfac, fdrgs, taup, k_g_s;

    /*寻找关于气体的脉络（初步的）*/
    /*寻找关于固体的脉络（第二相）*/
```

```
    thread_g = THREAD_SUB_THREAD(mix_thread, s_col);/*气体相*/
    thread_s = THREAD_SUB_THREAD(mix_thread, f_col);/*固体相*/

    /*寻找相速度和特性*/

    x_vel_g = C_U(cell, thread_g);
    y_vel_g = C_V(cell, thread_g);

    x_vel_s ＝ C_U(cell, thread_s);
    y_vel_s = C_V(cell, thread_s);

    slip_x = x_vel_g – x_vel_s;
    slip_y = y_vel_g – y_vel_s;

    rho_g = C_R(cell, thread_g);
    rho_g = C_R(cell, thread_s);

    mu_g = C_R(cell, thread_g)

    /*计算滑动*/
    abs_v = sqrt(slip_x*slip_x + slip_y*slip_y);

    /*计算 Reynold 数*/
    reyp = rho_g*abs_v*diam2/mu_g;

    /*计算粒子松弛时间*/
    taup ＝ rho_s*diam2*diam2/18./mu_g;

    void_g =C_VOF(cell, thread_g);/*气体体积组份*/

    /*计算阻力和返回阻力系数，k_g_s*/
    afac = pow(void_g, 4.14);

    if(void_g<=0.85)
      bfac = 0.281632*pow(void_g, 1.28);
    else
      bfac = pow(void_g, 9.076960);

    vfac = 0.5*(afac-0.06*reyp+sqrt(0.0036*reyp*reyp+0.12*reyp*
                  (2.*bfac-afac)+afac*afac));
    fdrgs = void_g*(pow((0.63*sqrt(reyp)/
                      vfac*4.8*sqrt(vfac)/vfac),2))/24.0;

    k_g_s = (1.-void_g)*rho_s*fdrgs/taup;

    return k_g_s;
}
```

3. DEFINE_ VECTOR_EXCHANGE_PROPERTY

使用 DEFINE_VECTOR_EXCHANGE_PROPERTY 宏可以为多相混合物模型规定用户滑动速度。

- 应用

宏：	DEFINE_VECTOR_EXCHANGE_PROPERTY (name, c, mixture_thread, second_column_phase_index, first_column_phase_index, vector_result)
自变量类型：	cell_t c Thread *mixture_thread int second_column_phase_index int first_column_phase_index real *vector_result
函数返回：	void

DEFINE_VECTOR_EXCHANGE_PROPERTY 中有六个自变量：name，c，mixture_thread，second_column_phase_index，first_column_phase_index 和 vector_result。各变量的解释与 DEFINE_EXCHANGE_PROPERTY 中的基本相同。这里 vector_result 也是被 FLUENT 的求解器传送到你的 UDF 的变量。UDF 将需要设置滑动速度矢量（例如，一个 2D 问题的 vector_result[0]，vector_result[1]）为滑动速度矢量（vector_result）的实型指示器所参考的值。

- 例子

下面名为 custom_slip 的 UDF，规定一两相混合物问题中的一个用户滑动速度。

```
/*******************************************************************
针对一个两相混合物问题，定义一个用户滑动速度的 UDF
*******************************************************************/
 #include “udf.h”

 DEFINE_VECTOR_EXCHANGE_PROPERTY(custom_slip, c, mixture_thread,
 second_column_phase_index, first_column_phase_index, vector_result)
 {
   real grav[2] = {0., -9.81};
   real K = 5.e4;

   real pgrad_x, pgrad_y;

   Thread *pt, *st;/*关于主要和次要相的脉络指示器*/

   pt = THREAD_SUB_THREAD(mixture_thread, second_column_phase_index);
   st = THREAD_SUB_THREAD(mixture_thread, first_column_phase_index)

   /*在这个点上，对于相的脉络，清楚的是主要相(0)和次要相(1)*/

   pgrad_x = C_DP(c, mixture_thread)[0];
   pgrad_y = C_DP(c, mixture_thread)[1];

   vector_result[0] = -(phrad_x/K)+(((C_R(c, st)-
                      C_R(c, pt))/K*grav[0]);
```

```
    vector_result[1] = -(phrad_y/K)+(((C_R(c, st)-
      C_R(c, pt))/K*grav[1]);
  }
```

18.2.5: 动态网格的 DEFINE 宏

应用本节所介绍的 DEFINE 宏，可以定义关于控制一个动态网格行为的 UDFs。值得大家注意的是，使用 DEFINE_CG_MOTION，DEFINE_GEOM 和 DEFINE_GRID_MOTION 定义的动态网格 UDFs，在 FLUENT 中仅可以作为汇编的 UDFs 被执行。表 18.2.5.1 给出了一个关于动态网格的 DEFINE 宏、DEFINE 宏定义的函数、以及 FLUENT 中被激活的面板的快捷参考指导。每一个 DEFINE 宏的定义都被放在了 udf.h 头文件中。为方便读者，把它们列在了附录 A 中。

此模型主要包括以下几种特殊的 DEFINE 宏：

- DEFINE_CG_MOTION
- DEFINE_GEOM
- DEFINE_GRID_MOTION

使用某个宏定义的 UDF，在被通译或汇编后，在 DEFINE 宏的自变量中规定的名字将会在 FLUENT 其对应的面板中变地可视。

表 18.2.5.1：关于动态特殊网格的 DEFINE 宏的快捷参考指导

函数	DEFINE 宏	激活的面板
重心移动	DEFINE_CG_MOTION	Dynamic Zones
网格移动	DEFINE_GRID_MOTION	Dynamic Zones
几何变形	DEFINE_GEOM	Dynamic Zones

1. DEFINE_CG_MOTION

当想通过由 FLUENT 提供的在每一时间步的线性和角速度，对一个特殊动态域的移动加以定义时，可以使用 DEFINE_CG_MOTION 宏。基于立体的运动，FLUENT 使用这些速度更新动态域上节点的位置。注意，使用 DEFINE_CG_MOTION 宏定义的 UDFs 仅可以作为汇编的 UDFs 被执行。

- 应用

宏：	DEFINE_CG_MOTION (name, dt, vel, omega, time, dtime)
自变量类型：	Dynamic_Thread *dt
	real vel[]
	real omega[]
	real time
	real dtime
函数返回：	void

DEFINE_CG_MOTION 中有六个自变量：name，dt，vel，omega，time 和 dtime。name 是用户给出的 UDF 的名字。dt，vel，omega，time 和 dtime 是被 FLUENT 的求解器传送到 UDF 的变量。线速度和角速度分别通过写入排列 vel 和 omega 被返回到 FLUENT。

dt 是结构的一个指示器，其中这个结构保存你所规定的或者 FLUENT 计算的动态网格分布。当前时间和时间步分别作为 time 和 dtime 由 FLUENT 给出。

- 例子

下面这个例子的线速度，是从 x 方向的体上的一个简单力平衡计算得到的，例如

$$\int_{t_0}^{t} dv = \int_{t_0}^{t} (F/m)dt \qquad (18.2.5.1)$$

其中，v 是速度，F 是力，m 是体质量。时间 t 的速度的计算使用的是一个显示 Euler 公式

$$v_t = v_{t-\Delta t} + (F/m)\Delta t \qquad (18.2.5.2)$$

```
/**********************************************************************
* 运动方程的一个自由度（x 方向）
* 汇编的 UDF
**********************************************************************/
 #include “udf.h”

 static real v_prev = 0.0;

 DEFINE_CG_MOTION(piston, dt, vel omega, time, dtime)
 {
   Thread *t;
   face_t f;
   real NV_VEC (A);
   real force, dv;

  /* 重新设置速度*/
   NV_S (vel, = , 0.0);
   NV_S (omega, =, 0.0);

  If (!Data_Valid_P ())
    return;

  /*获得脉络指示器，这个运动关于这个指示器而被定义*/
  t = DT_THREAD (dt);

  /*通过轮循过所有的面计算体压力*/
  force = 0.0;
  begin_f_loop (f, t)
   {
    F_AREA (A, f, t);
    Force += F_P (f, t) * NV_MAG (A);
   }
  end_f_loop (f,t)

  /*计算速度中的变化，如：dv=F*dt/mass,使用显示的 Euler 公式更新速度*/
  dv = dtime *force/50.0;
  v_prev += dv;
  Message (“time = %f, x_vel = %f, force = %f\n”, time, v_prev,
  Force);
```

```
  /*设置 x 速度分量*/
  vel[0] = v_prev;
}
```

2. DEFINE_GEOM

使用 DEFINE_GEOM 宏可以定义一个变形域的几何。默认的时候，FLUENT 给出了关于定义节点沿一个平面和圆柱面移动的一个机制。当 FLUENT 更新一个变形域上的一个节点的时候，通过调用 DEFINE_GEOM 宏来重新配置节点。注意，使用这里的宏定义的 UDFs 仅可以作为汇编的 UDFs 被执行。

- 应用

宏：	DEFINE_GEOM (name, d, dt, position)
自变量类型：	char name
	Domain *d
	Dynamic_Thread *dt
	real *position
函数返回：	void

DEFINE_GEOM 中有四个自变量：name，d，dt 和 position。name 是用户给出的 UDF 的名字。d，dt 和 position 是被 FLUENT 的求解器传送到 UDF 的变量。新位置通过写入位置的排列而返回到 FLUENT。

作为 DEFINE_CG_MOTION 宏，dt 是动态域的数据结构的一个指示器，d 是领域指示器。

- 例子

下面名为 parabola 的 UDF，作为一个汇编的 UDF 被执行。

```
/**********************************************************************
通过点(0, 1)， (1/2, 5/4)，(1, 1)，定义 parabola
**********************************************************************/
#include “udf.h”

DEFINE_GEOM(parabola, domain, dt, position)
{
  /*设 y = -x^2 + x + 1*/
  position[1] = - position[0]*position[0] + position[0] + 1;
}
```

3. DEFINE_GRID_MOTION

缺省时，FLUENT 通过应用固体移动方程更新一个动态域上的节点位置。这意味着在动态域上的节点之间没有相对移动。然而，如果需要独立地控制每一节点地移动，那么可以使用一个网格移动地 UDF。例如，关于一个 DEFINE_GRID_MOTION 的 UDF，在由于流体结构的交互而出现偏差的情况下，可以更新每个节点的位置。注意，使用这里的宏定义的 UDFs 仅可以作为汇编的 UDFs 被执行。

- 应用

宏：	DEFINE_GRID_MOTION (name, d, dt, time, dtime)
自变量类型：	char name
	Domain *d

Dynamic_Thread *dt

real time

real dtime

函数返回：　　void

DEFINE_GRID_MOTIN 中有五个自变量：name，d，dt，time 和 dtime。name 是用户给出的 UDF 的名字。d，dt，time 和 dtime 是被 FLUENT 的求解器传送到 UDF 的变量。

类似与上面的宏，dt 是动态网格结构的一个指示器，d 是领域指示器，time 是当前时间，dtime 是时间步。

● 例子

考虑下面例子，规定一个悬梁上的在 x 位置的偏差

$$\omega_y(x) = -10.4\sqrt{x}\sin 26.178t \qquad x > 0.02 \tag{18.2.5.3}$$

$$\omega_y(x) = 0 \qquad x <= 0.02 \tag{18.2.5.4}$$

其中，ω_y 是一个位置 x 上的角速度的 y 分量。节点位置被更新的基础是

$$(\vec{r})^{t+\Delta t} = (\vec{r})^t + \vec{\Omega} \times (\vec{r})^t \Delta t \tag{18.2.5.5}$$

其中，$\vec{\Omega}$ 是角速度，$\vec{r}$ 是动态域上一个节点的位置矢量。

```
/**********************************************************************
* 基于一个简单梁的偏差方程的节点移动
* 汇编的 UDF
**********************************************************************/
 #include “udf.h”

 DEFINE_GRID_MOTION(beam, domain, dt, time, dtime)
 {
  Thread *tf = DT_THREAD (dt);
  face_t f;
  Node *v;
  real NV_VEC (omega), NV_VEC (axis),NV_VEC (dx);
  real NV_VEC (origin), NV_VEC (rvec);
  real sign;
  int n;

  /*设邻近单元域上的变形标记*/
  SET_DEFORMING_THREAD_FLAG (THREAD_TO (tf));

  sign = -5.0 * sin (26.178 * time);

  Message (“time = %f, omega = %f\n”, time, sign);

  NV_S (omega, = , 0.0);
  NV_D (axis, =, 0.0, 1.0, 0.0);
  NV_D (origin, =, 0.0, 0.0, 0.152);
```

```
      begin_f_loop (f, tf)
       {
         f_node_loop (f, tf, n)
            {
              v = F_NODE (f, tf, n);

            /*如果 x 位置大于 0.02，以及当轮循过前面的面时，当前节点在前面没有被访问，
                那么更新节点 */
              if (NODE_X (v) > 0.021 && NODE_POS_NEED_UPDATE (v))
                 {
                   /*表明节点位置已被更新，并且只有一次更新*/

                   NODE_POS_UPDATED (v);
                   NV_VV (rvec, =, NODE_COORD (v), -, origin);
                   NV_CROSS (dx, omega, rvec);
                   NV_S (dx, *=, dtime);
                   NV_V (NODE_COORD (v), +=, dx);
                 }
           }
       }
       end_f_loop (f, tf);
     }
```

18.2.6: DPM DEFINE 宏

这节介绍了关于离散相模型（Discrete Phase Model，DPM）的 DEFINE 宏。表 18.2.6.1 给出了一个关于 DPM DEFINE 宏、宏定义的函数、以及 FLUENT 中被激活的面板的快捷参考指导。每一个 DEFINE 宏的定义都被放在了 udf.h 头文件中。为方便读者，把它们列在了附录 A 中。

主要包括以下几种特殊的 DEFINE 宏：

- DEFINE_DPM_BC
- DEFINE_DPM_BODY_FORCE
- DEFINE_DPM_DRAG
- DEFINE_DPM_EROSION
- DEFINE_DPM_INJECTION_INIT
- DEFINE_DPM_LAW
- DEFINE_DPM_OUTPUT
- DEFINE_DPM_PROPERTY
- DEFINE_DPM_SCALAR_UPDATE
- DEFINE_DPM_SOURCE
- DEFINE_DPM_SPRAY_COLLIDE
- DEFINE_DPM_SWITCH

使用某个宏定义的 UDF，在被通译或汇编后，在 DEFINE 宏的自变量中规定的名字将会在 FLUENT 其对应的面板中变地可视。

表 18.2.6.1：关于特殊 DPM DEFINE 宏的快捷参考指导

函数	DEFINE 宏	激活的面板
边界上的粒子状态	DEFINE_DPM_BC	boundary condition (例如，Velocity Inlet)
粒子上的体力	DEFINE_DPM_BODY_FORCE	Discrete Phase Model
粒子和流体之间的阻力系数	DEFINE_DPM_DRAG	Discrete Phase Model
腐蚀和增长率	DEFINE_DPM_EROSION	Discrete Phase Model
初始化注射	DEFINE_DPM_INJECTION_INIT	Set Injection Properties
关于粒子的用户法则	DEFINE_DPM_LAW	Custom Laws
写入取样平面输出的修改	DEFINE_DPM_OUTPUT	Sample Trajectories
材料特性	DEFINE_DPM_PROPERTY	Materials
每次粒子位置被更新时的更新标量	DEFINE_DPM_SCALAR_UPDATE	Discrete Phase Model
粒子源项	DEFINE_DPM_SOURCE	Discrete Phase Model
粒子碰撞运算法则	DEFINE_DPM_SPRAY_COLLIDE	User-Defined Function Hooks
关于法则间的转换的改变标准	DEFINE_DPM_SWITCH	Custom Laws

1. DEFINE_DPM_BC

如果想定义关于粒子的边界条件，那么可以使用 DEFINE_DPM_BC 宏。每次一个粒子接触域的边界时函数被执行，除了对称或周期边界。可以使用 DEFINE_DPM_BC 对每一边界定义一个离散的 UDF。

- 应用

宏：	DEFINE_DPM_BC (name, p, t, f, f_normal, dim)
自变量类型：	Tracked_Particle *p
	Thread *t
	face_t f
	real f_normal[]
	int dim
函数返回：	int

DEFINE_DPM_BC 中有六个自变量：name，p，t，f，f_normal 和 dim。name 是用户给出的 UDF 的名字。p，t，f，f_normal 和 dim 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要计算一个粒子撞击壁面之后的新速度，然后返回粒子轨迹的状态（作为一个整型）。

被传送的变量 p 是 Tracked_Particle 数据结构的一个指示器，其中这个数据结构包含与所跟踪的粒子相关的数据。这个指示器可以作为一个特殊粒子宏的自变量使用，从而获得有关粒子特性的资料。t 是粒子当前撞击的面脉络的一个指示器。f 是一个指数，它被用来鉴别粒子撞击着的面。f_normal 是一个包含面的法线单位矢量的实型排列。dim 是流动问题的维数。对于二维，二维轴对称以及二维轴对称漩涡的流动，维数的值均为 2，对于三维流动，维数的值是 3。

2. DEFINE_DPM_BODY_FORCE

使用DEFINE__DPM_BODY_FORCE宏可以规定除了粒子上的重力或阻力以外的体力。

- 应用

宏：	DEFINE_DPM_BODY_FORCE (name, p, i)
自变量类型：	Tracked_Particle *p
	int i
函数返回：	real

DEFINE_DPM_BODY_FORCE 中有三个自变量：name，p 和 i。name 是用户给出的 UDF 的名字。p 和 i 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要把由体力产生的加速度的实型值返回到 FLUETN 求解器。

被传送的变量 p 是 Tracked_Particle 数据结构的一个指示器，其中这个数据结构包含与所跟踪的粒子相关的数据。这个指示器可以作为一个特殊粒子宏的自变量使用，从而获得有关粒子特性的信息，例如注射特性。i 是取值为 0，1 和 2 的一个指数，它被用来区别被函数所返回的笛卡儿（Cartesian）体力分量。

3. DEFINE_DPM_DRAG

使用 DEFINE_DPM_DRAG 宏能够规定粒子和流体之间的阻力系数，C_D，通过下面的方程定义：

$$F_D = \frac{18\mu}{\rho_p D_p^2} \frac{C_D \operatorname{Re}}{24}$$

- 应用

宏：	DEFINE_DPM_DRAG (name, Re, p)
自变量类型：	real Re
	Tracked_Particle *p
函数返回：	real

DEFINE_DPM_DRAG 中有三个自变量：name，Re 和 p。name 是用户给出的 UDF 的名字。Re 和 p 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要把一个粒子上阻力的实型值返回到求解器，这个值必须是无量纲的，其表示为$18 * Cd * \operatorname{Re}/ 24$。

被传送的变量 Re 是特征速度为相对气体速度，特征尺度为粒子直径的粒子 Reynolds 数。p 是 Tracked_Particle 数据结构的一个指示器，其中这个数据结构包含与所跟踪的粒子相关的数据。这个指示器可以作为一个特殊粒子宏的自变量使用，从而获得有关粒子特性的信息。

4. DEFINE_DPM_EROSION

使用 DEFINE_DPM_BC 宏能够接近作为粒子流撞击一个壁面表面计算得到的腐蚀和增长率。当粒子遇到一个反射表面的时候，函数被访问。

- 应用

宏：	DEFINE_DPM_BC (name, p, t, f, normal, alpha, Vmag, mdot)
自变量类型：	Tracked_Particle *p
	Thread *t
	face_t f
	real normal[]
	real alpha
	real Vmag
	real mdot
函数返回：	void

DEFINE_DPM_EROSION 中有八个自变量：name，p，t，f，normal，alpha，Vmag 和 mdot。name 是用户给出的 UDF 的名字。p，t，f，normal，alpha，Vmag 和 mdot 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要计算关于腐蚀率或增长率的值，存储 F_STORAGE_R（f, t, SV_DPMS_EROSION）和 F_STORAGE_R（f, t, SV_DPMS_ACCRETION）里的面上的值。

被传送的变量 p 是 Tracked_Particle 数据结构的一个指示器，其中这个数据结构包含与所跟踪的粒子相关的数据。这个指示器可以作为一个特殊粒子宏的自变量使用，从而获得有关粒子特性的信息。t 是粒子当前撞击的面脉络的一个指示器。f 是一个指数，它被用来鉴别粒子撞击着的面。normal 是一个包含面的法线单位矢量的实型排列。alpha 是一个表示粒子速度大小的实型变量（单位为 m/s）。mdot 是粒子流撞击面时的实型流动比率（单位为 kg/s）。

5. DEFINE_DPM_INJECTION_INIT

使用 DEFINE_DPM_INJECTION_INIT 宏能够对一个粒子的注射特性（例如特定区域，直径和速度）进行初始化。

- 应用

宏：	DEFINE_DPM_ INJECTION_INIT(name, I)
自变量类型：	Injection *I
函数返回：	void

DEFINE_DPM_INJECTION_INIT 中有两个自变量：name 和 I。name 是用户给出的 UDF 的名字。I 是被 FLUENT 的求解器传送到 UDF 的变量。

被传送的变量 I 是 Injection 结构的一个指示器，其中这个结构里包含正在产生的粒子。关于每个 Injection，在第一个 DPM 反复之前，函数被访问两次，然后在粒子注射进后来每一 DPM 反复的域里之前，函数在被访问一次。

6. DEFINE_DPM_LAW

如果有意用户化粒子的法则，那么使用 DEFINE_DPM_LAW 宏。例如，使用这个宏而建立的 UDF 能够对小滴和燃烧粒子的热传递和质量传递的速率规定用户法则。除此之外，可以在小滴或粒子与其周围进行质量或能量交换的时候，规定关于质量，直径和温度特征的用户法则。

- 应用

宏：	DEFINE_DPM_ LAW(name, p, ci)
自变量类型：	Tracked_Particle *p
	int ci

函数返回：　　　　void

DEFINE_DPM_LAW 中有三个自变量：name，p 和 ci。name 是用户给出的 UDF 的名字。p 和 ci 是被 FLUENT 的求解器传送到 UDF 的变量。

被传送的变量 p 是 Tracked_Particle 数据结构的一个指示器，其中这个数据结构包含与所跟踪的粒子相关的数据。这个指示器可以作为一个特殊粒子宏的自变量使用，从而获得有关粒子特性的信息。ci 是一个整数，它表示连续相和离散相是否耦合（若连续相耦合，则等于 1，不耦合则为 0）。

7. DEFINE_DPM_OUTPUT

当有意修改所写入取样装置输出里的内容时，可以使用 DEFINE_DPM_OUTPUT 宏。所定义的 UDF 允许在一个粒子经过一个取样器时存取被书写的变量。

- 应用

宏：	DEFINE_DPM_ OUTPUT(name, header, fp, p, t, plane)
自变量类型：	int header
	FILE *fp
	Tracked_Particle *p
	Thread *t
	Plane *plane
函数返回：	void

DEFINE_DPM_OUTPUT 中有六个自变量：name，header，fp，p，t 和 plane。name 是用户给出的 UDF 的名字。header，fp，p，t 和 plane 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 的输出将被写入一个由 fp 指出的文件里。

被传送的变量 header 是一个整数，在粒子被跟踪之前，函数第一次访问时，这个整数等于 1，在以后的访问中被设为 0。fp 是写入或读出文件的一个指示器。p 是 Tracked_Particle 数据结构的一个指示器，其中这个数据结构包含与所跟踪的粒子相关的数据。这个指示器可以作为一个特殊粒子宏的自变量使用，从而获得有关粒子特性的信息。

t 是脉络的一个指示器，如果取样器由一个网格表面所表示，那么粒子经过这个脉络。如果取样器没有作为一个网格表面被定义，那么 t 的值是 NULL。若取样装置作为一个平面的一部分（在 2D 中是线）被定义，则 plane 表示 Plane 结构的一个指示器。如果一个网格表面被取样器所使用，那么 plane 的值是 NULL。

8. DEFINE_DPM_PROPERTY

当用户有意指定离散相材料的特性时，可以使用 DEFINE_DPM_PROPERTY 宏。使用这个宏可以定义下面所列出的离散相特性：

- particle emissivity
- vapor pressure
- vaporization temperature
- particle scattering factor
- boiling point
- particle viscosity
- particle surface tension

- 应用

宏：	DEFINE_DPM_ PROPERTY(name, c, t, p)
自变量类型：	cell_t c
	Thread *t

Tracked_Particle *p

函数返回：　　real

DEFINE_DPM_PROPERTY 中有四个自变量：name，c，t 和 p。DEFINE_DPM_PROPERTY 除了增加了 Tracked_Particle p 的指示器外，与 DEFINE_ PROPERTY 的函数有相同的自变量。name 是用户给出的 UDF 的名字。c，t 和 p 是被 FLUENT 的求解器传送到 UDF 的变量。UDF 将需要计算离散相特性的实型值，并把这个值返回到求解器。

被传送的变量 c 是一个指数，它被用来区别所给脉络中粒子所处的单元。c 是粒子所处的脉络的一个指示器。p 是 Tracked_Particle 数据结构的一个指示器，其中这个数据结构包含与所跟踪的粒子相关的数据。这个指示器可以作为一个特殊粒子宏的自变量使用，从而获得有关粒子特性的信息。

9. DEFINE_DPM_SCALAR_UPDATE

使用 DEFINE_DPM_SCALAR_UPDATE 宏能够在每次粒子的位置被更新时更新标量。函数允许与粒子相关的变量在粒子存在的整个过程中被更新或积分。粒子的值可以保存在与 Tracked_Particle 相关的一个数列中。计算和保存在这个数列中的值可以用来改变粒子轨迹的颜色。

- 在 FLUENT 执行过程中，DEFINE_DPM_SCALAR_UPDATE 的函数在粒子开始积分的时候（当 initialize 等于 1 时）被访问，之后，在关于粒子轨迹积分的每一时间步之后被访问。

- 应用

宏：　　DEFINE_DPM_SCALAR_UPDATE(name, c, t, initialize, p)

自变量类型：　　cell_t c
Thread *t
int initialize
Tracked_Particle *p

函数返回：　　void

DEFINE_DPM_SCALAR_UPDATE 中有五个自变量：name，c，t，initialize 和 p。name 是用户给出的 UDF 的名字。c，t，initialize 和 p 是被 FLUENT 的求解器传送到 UDF 的变量。

被传送的变量 c 是一个指数，它被用来区别粒子当前所处的单元。t 是粒子当前所处的脉络的一个指示器。p 是 Tracked_Particle 数据结构的一个指示器，其中这个数据结构包含与所跟踪的粒子相关的数据。这个指示器可以作为一个特殊粒子宏的自变量使用，从而获得有关粒子特性的信息。实型数列 user 适用于存储。这个数列的大小应当在 **Number of Scalars** 领域里的 **Discrete Phase Model** 面板中给出设置。

10. DEFINE_DPM_SOURCE

使用 DEFINE_DPM_SOURCE 宏能够存取粒子的源项。对于一个所给单元里的一个粒子，在积聚的源项被添加到关于耦合 DPM 计算的质量、能量和动量交换相之前，函数允许使用这些源项。

- 应用

宏：　　DEFINE_DPM_SOURCE(name, c, t, S, strength, p)

自变量类型：　　cell_t c
Thread *t
dpms_t *S
real strength
Tracked_Particle *p

函数返回：　　　　void

DEFINE_DPM_SOURCE 中有六个自变量：name，c，t，S，strength 和 p。name 是用户给出的 UDF 的名字。c，t，S，strength 和 p 是被 FLUENT 的求解器传送到 UDF 的变量。

被传送的变量 c 是一个指数，它被用来区别粒子当前所处的单元。t 是粒子当前所处在的脉络的一个指示器。S 是源结构 dpms_t 的一个指示器，这个源结构包含单元的源项。strength 是粒子数流动速率。p 是 Tracked_Particle 数据结构的一个指示器，其中这个数据结构包含与所跟踪的粒子相关的数据。这个指示器可以作为一个特殊的粒子宏的自变量使用，从而获得有关粒子特性的信息。一旦修改的源项被函数所计算，它将被存储在 S 中。

11. DEFINE_DPM_SPRAY_COLLIDE

可以使用 DEFINE_DPM_SPRAY_COLLIDE 宏避开 FLUENT 缺省的喷雾碰撞的运算法则。当小滴互相碰撞时，它们也许会反跳，在这种情况下小滴的速度改变；它们也许会合而为一，在这种情况下它们在 **DPM** 面板里的速度、直径以及数目都会改变。一个喷雾碰撞的 UDF 在每一小滴时间步之后的小滴跟踪的过程中被访问，并且 UDF 要求 **DPM** 面板中的碰撞是激活的。

- 应用

宏：　　　　D EFINE_DPM_SPRAY_COLLIDE(name, tp, p)
自变量类型：　　Tracked_Particle *p
　　　　　　　　Particle *p
函数返回：　　　void

DEFINE_DPM_SPRAY_COLLIDE 中有三个自变量：name，tp 和 p。name 是用户给出的 UDF 的名字。tp 和 p 是被 FLUENT 的求解器传送到 UDF 的变量。

输入的变量 p 是 tp，Tracked_Particle 结构变量，和 p，Particle 结构变量。FLUENT 把粒子 p 存储在一个连接的列表中。当碰撞被激活的时候，这个连接的列按粒子当前所处在的单元的顺序排列的。粒子从连接的列表中被跟踪时，被从粒子列表中复制到了一个 Tracked_Particle 结构里。

12. DEFINE_DPM_SWITCH

可以使用 DEFINE_DPM_SWITCH 宏修改法则之间转换的标准。函数可以用来控制用户定义的粒子法则和默认的粒子法则之间，或者不同的用户定义法则之间，或者默认的粒子法则之间的转换。

- 应用

宏：　　　　D EFINE_DPM_SWITH(name, p, ci)
自变量类型：　　Tracked_Particle *p
　　　　　　　　int ci
函数返回：　　　void

DEFINE_DPM_SWITCH 中有三个自变量：name，p 和 ci。name 是用户给出的 UDF 的名字。p 和 ci 是被 FLUENT 的求解器传送到 UDF 的变量。

p 是 Tracked_Particle 数据结构的一个指示器，其中这个数据结构包含与所跟踪的粒子相关的数据。这个指示器可以作为一个特殊粒子宏的自变量使用，从而获得有关粒子特性的信息。ci 是一个整数，它指出了连续和离散相是否耦合，若连续相耦合，则它等于 1，不耦合则为 0。

18.3　通译和编译及连接用户定义函数（UDF）

一旦使用文本编辑器写了 UDF，并且以扩展名 a.c 的形式把源文件保存在用户当前的工作目录中，那么就要准备对其进行通译或编译以及装载。而通译或编译完 UDF 之后，需要把它连接到 FLUENT 中，并且在用户的 FLUENT 模型中使用这个函数。我们将从以下 8 小节具体的介绍相关内容：

18.3.1 关于通译和编译用户定义函数的概述
18.3.2 使用 **Interpreted UDFs** 面板通译一个源文件
18.3.3 使用 **Compiled UDFs** 面板编译一个源文件
18.3.4 连接一般求解器的 UDFs 到 FLUENT
18.3.5 连接物种模型的 UDFs 到 FLUENT
18.3.6 连接多相的 UDFs 到 FLUENT
18.3.7 连接动态网格的 UDFs 到 FLUENT
18.3.8 连接 DPM UDFs 到 FLUENT

18.3.1　关于通译和编译用户定义函数的概述

通译和编译的英文解释分别为 interprete 和 compile。在这里，分别称通译和编译之后的 UDF 为通译的 UDF 和编译的 UDF。包含 UDFs 的源代码文件在 FLUENT 中既可以被通译也可以被编译，但通译与编译 UDF 也有不同之处。

1. 通译的 UDFs

通译的 UDFs 是在运行时直接从一个源文件（例如，udfexample.c）通译的。其过程包括一个到 **Interpreted UDFs**（通译的 UDFs）面板的访问，在这个面板上，用户可以在一单独步骤中，在源文件（例如，udfexample.c）里通译相应的 UDF。一旦通译了一个源文件，不妨把 C 函数的名字及内容保存在 case 文件里，即写 case 文件。这样做的好处在于，无论今后何时读进这个 case 文件，所定义的函数都将被自动地通译。UDFs 或者通过 **Interpreted UDFs** 面板手动地通译，或者通过读进一个 case 文件而被自动地通译，而一旦被通译，所有包含在源文件里的通译的 UDFs 将在 FLUENT 的图形用户界面的面板中变地可视和可选。

在 FLUENT 的内部，使用一个 C 语言预处理程序，源代码被编译成一个中间的、体系机构独立的机码。当 UDF 被调用时，这个机码在一个内部仿真程序装置或注释器上执行。额外层的代码导致一个执行出现失败，但它允许在不同的体系机构、运行体系和 FLUENT 版本之间共享一个通译的 UDF。如果执行速度不成问题，那么一个通译的 UDF 总是能够不加修改的在编译的模式中被运行。

2. 编译的 UDFs

关于编译的 UDFs，其编辑过程包括两步：建立和装载。首先，访问 **Compiled UDFs**（汇编的 UDFs）面板，在这个面板上从一个或多个源文件 **Build**（建立）一个共享库，然后把共享库（例如，libudf）**Load**（装载）进 FLUENT 中。一旦共享库被装载，可以把它写进 case 文件中，以便今后读进这个 case 文件的时候，共享库被自动地装载。这避免了每次运行一个模拟时必须重新装载所编译的库。UDFs 或者通过 **Compiled UDFs** 面板手动地编译，或者通过读进一个 case 文件而被自动地编译，而一旦被编译，所有包含在共享库里的

编译的 UDFs 将在 FLUENT 的图形用户界面的面板中变地可视和可选。

在 FLUENT 内部，称作 Makefile 的原本被用来调用 C 系统编译器去建立一个目标代码库，其中这个目标代码库包含关于用户的 C 源代码的本地机语言翻译。这个共享库（例如，libudf）在运行时被装载进 FLUENT，或者读进一 case 文件时自动地装载，装载是通过一个称作“动态装载”（”dynamic loading”）的过程完成的。对于所用的计算机体系机构和可执行的 FLUENT 的特殊版本而言，目标库都是特定的。所以，每更新 FLUEN 时，都必须要重新建立库，其中计算机操作系统标准的改变，运行工作所用机器的类型的改变，都属于 FLUENT 的更新。

3. udf.h 文件的位置

使用 DEFINE 宏定义的 UDFs（请看第 18.2 节）以及关于 DEFINE 宏的定义都包含在 udf.h 头文件里。所以，在通译或编译自己的源文件之前，udf.h 文件将需要在自己的工作路径中存取和保存。

udf.h 文件的位置为：

path/Fluent.Inc/fluent6.x/src/udf.h

其中，*path* 是用户自己安置好的释放目录，Fluent.Inc，所在的目录，替代 x 的是用户所释放的对应数字（如，fluent6.1 的 1）。

！一般说来，不应当从安装区域复制 udf.h 文件。策划编译器先从当前目录中寻找这个文件，如果在当前目录中找不到，编译器将自动地在/src 目录中寻找。在更新释放区域，但不从工作目录中移走一个旧版本的 udf.h 的情况下，不能访问这个文件的最新版本。

！在任何情况下，都不应当改变这个 udf.h 文件。

18.3.2 使用 Interpreted UDFs 面板通译一个源文件

本部分将介绍在 FLUENT 中如何通译一个源文件。UDFs 一旦被通译，包含在源文件里的 UDFs 的名字将出现在图形用户界面面板的相关列表中。

通译一个源文件的一般步骤如下：

1. 确定 UDF 源文件所在的目录与 case 和 data 文件所在目录相同。

   注意，如果你在 Windows 机器的一个网络上运行 FLUENT 的并行版本，那么你必须“共享”这个含有你的 UDF 源文件，case 文件和 data 文件的工作目录，以便簇中的所有计算节点能够看到这个目录。要进行此工作：

   打开 **Windows Explorer**（Windows 探测器或探针）的应用程序，正确地点击关于工作目录的文件夹，选择 **Sharing…**（共享）选项，并指定一个 **Share Name**（共享名字）（例如，mywork）。

2. 从你的工作目录中启动 FLUENT。
3. 读进（或建立）你的 case 文件。
4. 打开 **Interpreted UDFs** 面板（如图 18.3.2.1）

Define → User-Defined → Functions → Interpreted…

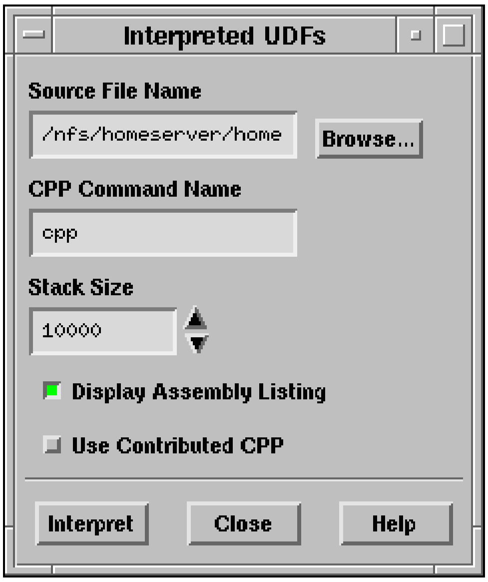


图 18.3.2.1：Interpreted UDFs 面板

5. 在 **Interpreted UDFs** 面板中，或者在 **Source File Name**（源文件的名字）下面的文本框里键入完全路径，或者使用浏览器选择你要通译的 UDFs 源文件。
   a. 使用浏览器选择你的源文件，在 **Interpreted UDFs** 面板中的 **Source Files**（源文件）下点击 **Browse…**（浏览），这样将会打开 **Select File**（选择文件）面板（如图 18.4.2.2 所示）。

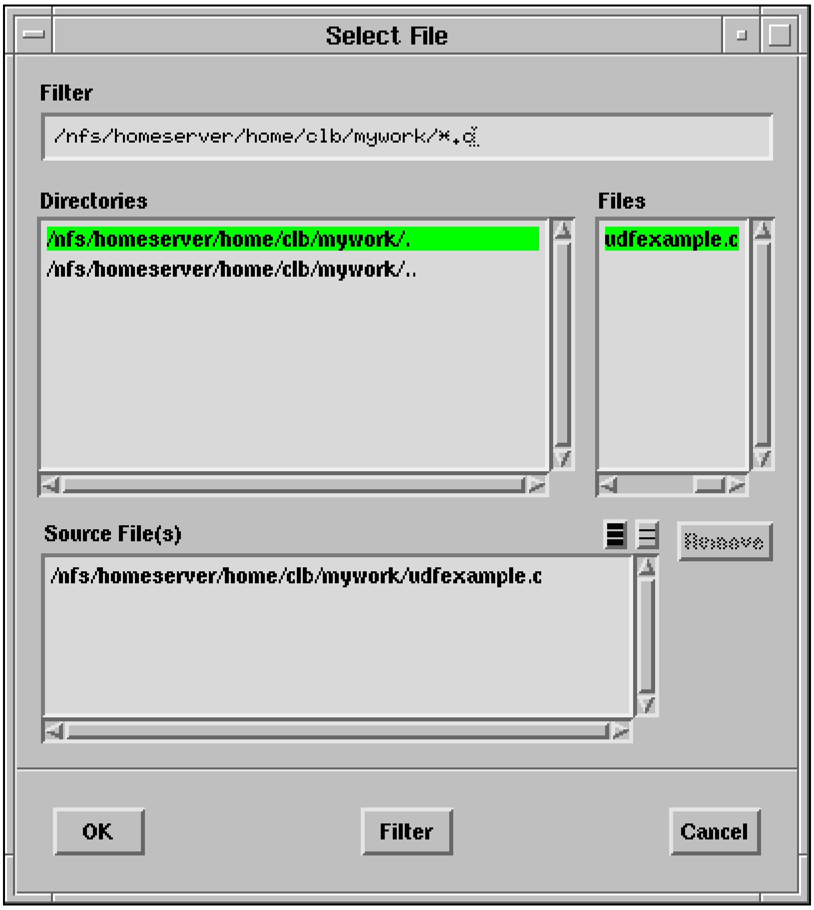


图 18.3.2.2：Select File 面板

b. 在 **Select File** 面板中，加亮 **Directories**（目录）下面框中的目录（例如，/nfs/homeserver/home/clb/mywork/）和 **Files**（文件）下面框中的你所需要的文件（如，udfexample.c），然后点击 OK。

！如果你在 Windows 机器的一个网络上运行 FLUENT，那么也许你需要在 **Source File Name** 下面的文本框中键入文件的完全路径，而不是使用浏览器选项。例如，通译一个放在名为 mywork 的一个共享工作目录里的文件，udfexample.c，那么你要在 **Source File Name** 下面的文本框中输入：

\\<fileserver>\mywork\udfexample.c

取代<fileserver>的是你的工作目录（mywork）和源文件(udfexample.c)所在的机器名。

6. 在 **Interpreted UDFs** 面板中，指定要被用于 **CPP Command Name**（CPP 命令名）下面框中的 C 预处理程序。你可以保留缺省的 cpp，或者选择 **Use Contributed CPP**（使用有贡献的 CPP）以便使用由 Fluent Inc 所给出的预处理程序。

   ！如果你从“PrePost”CD 盘安置了/contrib 成分，那么出现在面板中的将时默认的 cpp 预处理程序。对于 Windows NT 用户来说，FLUENT 的 Windows NT 安装中包含 cpp 预处理程序。

7. 保留 **Stack Size**（堆叠大小）里的默认值 10000，除非你函数里的局部变量数目会致使堆叠溢出。若发生堆叠溢出，在 **Stack Size** 下面的框中设置一个比所使用的局部变量大的数。

8. 如果在函数通译时，你要求一列汇编语言代码出现在你的控制台窗口里，那么保持 **Display Assembly Listing**（显示汇编列）选项。这个选项将被保存在你的 case 文件中，以便当你在以后读这个 case 文件时，汇编代码会自动地显示。

9. 点击 **Interpret**（通译）通译你的 UDF。

```
inlet_x_velocity:
                .local.pointer thread (r0)
                .local.int nv (r1)
    0           .local.end
    0           save
                .local.int f (r3)
    1           push.int 0
                .local.pointer x (r4)
    3           begin.data 8 bytes, 0 bytes initialized:
    7           save
    .             .
    .             .
    .             .
    156          pre.inc.int f (r3)
    158          pop.int
    159          b   .L3 (22)
         .L2:
    161          restore
    162          restore
    163          ret.v
```

！注意，如果编辑失败，那么 FLUENT 将报告一个错误，你需要调试你地程序。

10．通译完成后，关闭 Interpreted UDFs 面板。

11．如果想把通译地函数保存到 case 文件中，那么写 case 文件，在以后读这个文件时，函数会被自动地通译。如果选择了 **Display Assembly Listing** 选项，那么汇编代码将会出现在控制台的窗口里。

通译你的源文件时，常见错误

当你通译你的源文件时，会出现一个常见的错误。如果在你通译 UDF 时出现了编辑错误，那么这些错误将出现在控制台窗口中。然而，如果屏幕滚动过快，也许你就看不到所有的错误信息。就这个原因而言，你可能想在调试你的 UDF 时借助 **Display Assembly Listing** 选项。你可以从"log"文件中查看编辑的历史记录，这个文件在你的工作目录中。

如果你在调试你的 UDF 的过程中，Interpreted UDFs 面板一直是打开的，那么，自从你在一个隔开的窗口用编辑器做修改以来，Interpret 按钮可以被反复的使用。然后，你可以继续调试和通译你的 UDF，直到没有一个错误报告。记得在企图再通译之前，保存你再编辑窗口中对源代码文件所做的修改。

在通译源文件时，会出现的其中一个常见错误是企图通译含有解释程序所不供给的 C 语言要素的一些源代码。例如，如果你拥有包含一个构成参考命令（它不被 C 预处理程序支持）的代码，那么通译将会失败，你也将得到一个错误信息：

Error: /nsf/erflnx/home/erf/fluent/udfexample.c:

line 15: structure reference

关于并行 FLUENT 的特殊考虑

如果你在一个 Windows 网络上运行 FLUENT 的并行版本，当你试图通译你的源文件时遇到了错误，那么这可能是由安装 cpp 时不正确所导致的。合理的安装 FLUENT 的并行程序保证了 FLUENT_INT 环境变量被放置在共享目录中，FLUENT 就被装在这个目录中。如果变量被局部地定义，当你试图通译一个源文件时将会报告下面的错误信息：

Warning: unable to run cpp

18.3.3 使用 Compiled UDFs 面板编译一个源文件

编译源文件的步骤包括两步：建立和装载。对于源文件，首先建立一个共享库，然后把这个库装载到 FLUENT 里。一旦共享库被装载，在 **Compiled UDFs**（汇编的 UDFs）面板上你首次 **Build**（建立）了一个来自一个或多个源文件中的共享库，步骤包括对这个面板的访问，然后把共享库 **Load**（装载）进 FLUENT 中。一旦被装载，含在共享库中的编译 UDFs 将出现在 FLUENT 的图形用户界面面板里的相关列表中。

编译源文件的一般步骤如下：

1．确定 UDF 源文件所在的目录与你的 case 和 data 文件所在目录相同。

   注意，如果你在 Windows 机器的一个网络上运行 FLUENT 的并行版本，那么你必须"共享"这个含有你的 UDF 源文件，case 文件和 data 文件的工作目录，以便簇中的所有计算节点能够看到这个目录。要进行此工作：

      打开 **Windows Explorer**（Windows 资源管理器）的应用程序，正确地点击关于工作目录的文件夹，选择 **Sharing…**（共享）选项，并指定一个 **Share Name**（共享名字）（例如，mywork）。

2．从你的工作目录中启动 FLUENT。

3．读进（或建立）你的 case 文件。

4．打开 **Compiled UDFs** 面板（如图 18.3.3.1 所示）

Define → User-Defined → Functions → Compiled…

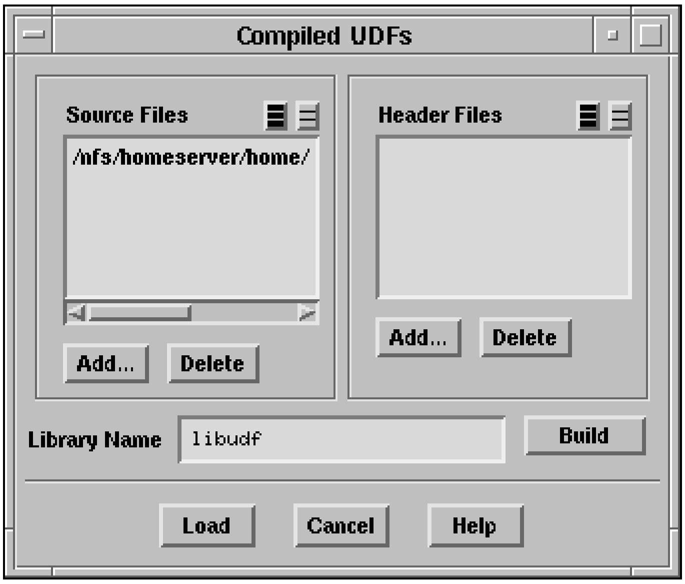


图 18.3.3.1：Compiled UDFs 面板

5．在 **Compiled UDFs** 面板中的 **Header Files** 下点击 **Add…**，选择你要编译的 UDF 头文件（如图 18.3.3.1）。这样将会打开 **Select File** 面板。

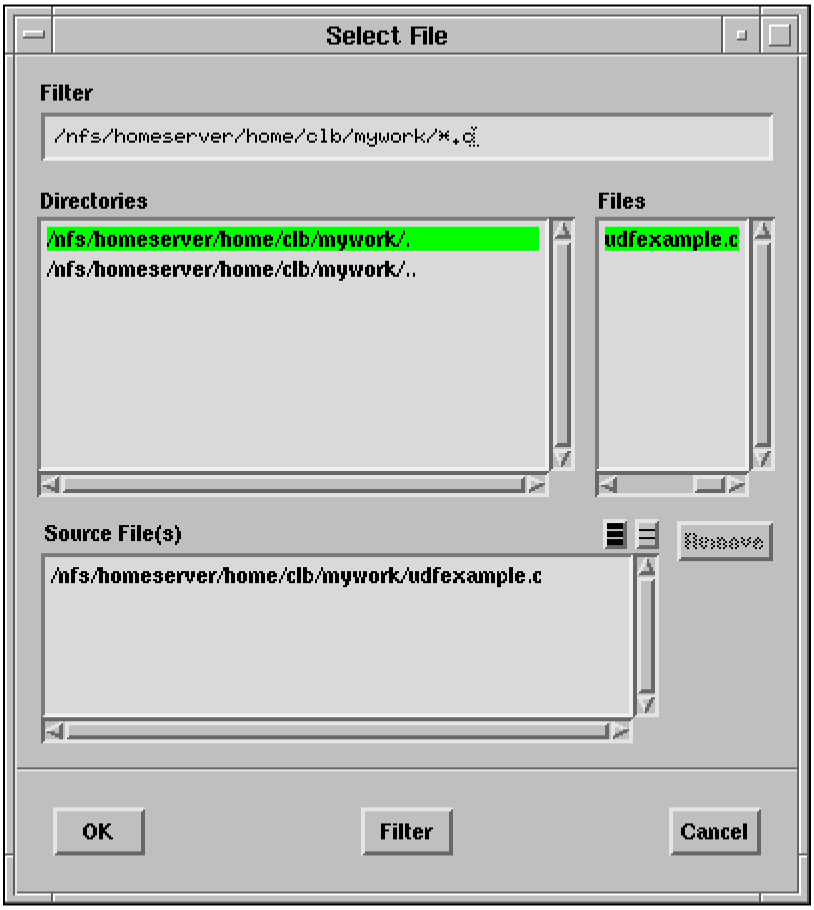


图 18.3.3.2：Select File 面板

a. 在 **Select File** 面板（如图 18.3.3.2 所示）中，加亮 **Directories** 下面框中的目录和 **Files** 下面框中的你所需要的文件（如，udfexample.c）。一旦被加亮，源文件的完全路径将在 **Source Files** 下面的框中显示出。点击 OK。

重复此步，直到选取了所有的源文件。

！如果你在 Windows 机器的一个网络上运行 FLUENT，那么也许你需要在 **Source File Name** 下面的文本框中键入文件的完全路径，而不是使用浏览器选项。例如，编译一个放在名为 mywork 的一个共享工作目录里的文件，udfexample.c，那么你要在 **Source File Name** 下面的文本框中输入：

\\<fileserver>\mywork\udfexample.c

取代<fileserver>的是你的工作目录（mywork）和源文件(udfexample.c)所在的机器名字。

6．在 **Compiled UDFs** 面板中的 **Source Files** 下点击 **Add…**（添加），选择你要编译的 UDF 源文件。这样将会打开 **Select File** 面板（如图 18.3.3.2 所示）。

a. 在 **Select File** 面板（如图 18.3.3.2 所示）中，加亮 **Directories** 下面框中的目录和 **Files** 下面框中的你所需要的文件。一旦被加亮，头文件的完全路径将在 **Header Files** 下面的框中显示出。点击 OK。

重复此步，直到选取了所有的头文件。

7. 在 **Compiled UDFs** 面板中，为你的源文件建立一个共享库的方法是在 **Libarary Name** 右边的文本框里键入共享库的名字，或者保留默认的名字 libudf，然后点击 **Build**。当建立过程开始时，会出现一个 **Information**（信息）对话框（如图 18.3.3.3 所示），建立的结果会显示在这个控制台窗口中。点击 OK 关掉此对话框。你可以在‘log’

文件中查看编辑历史，这个‘log’文件存放在你的工作目录下。

！如果你的建立失败，那么 FLUENT 将会把错误报告出，你将需要调试你的程序，然后才可能继续进行工作。

8．如果你没有再一个 Windows 网络上使用 FLUENT 的并行版本，那么通过点击 **Load** 来装载你刚刚建立的共享库。在一个控制台窗口上将显示一条信息，信息给出了装载过程的一个情形。例如：

Opening library “libudf”…

Library “libudf/hpux11/2d/libudf.so” opened

inlet_x_velocity

Done.

！如果你在 Windows 机器的一个网络上运行 FLUENT 的并行版本，请按照下面的方式做：

\\<fileserver>\mywork\udfexample.c

a.　在 **Compiled UDFs** 面板中，在 **Library Name** 的相应框中键入 \\<fileserver>\mywork\libudf

取代<fileserver>的是你的工作目录（mywork）和共享库的目录 (libudf)所在的机器名。点击 **Load**。

9．当你的库装载完之后，关掉 Compiled UDFs 面板。

10．写 case 文件。这样共享库中的编译函数就被保存在了这个 case 文件中，当你以后读进这个文件的时候，编译的函数会被自动地装载进 FLUENT 里。

！如果没把共享库保存到 case 文件中，那么以后再读进这个 case 文件并且用到此共享库时，必须使用 **Compiled UDFs** 面板把它装载进 FLUENT。

装载你的编译库时，常见错误

如果在一个 case 文件中保存了一个共享库，而后来你把这个库移动到了一个与起初不同的位置，那么当你读进这个 case 文件时，你会得到一个错误信息。

Opening library “libudf…

Error: open_udf_library: couldn’t open library:libudf/ln86/2d/libudf.so

在一个共享库被建立之前，当你企图装载它时，你会得到一个错误信息。

Opening library “libudf…

Error: open_udf_library: couldn’t open library:No such file or directory

关于并行 FLUENT 的特殊考虑

如果你在一个 Windows 网络上运行 FLUENT 的并行版本，包含共享库（例如，sharedlibudf）的目录必须包含簇中的所有远程节点。如果你忘记使用 Windows Explorer 激活库所在目录的共享选项，那么当你试图在 **Compiled UDFs** 面板中装载库（例如，sharedlibudf）时，FLUENT 将暂停工作。

18.3.4　连接一般求解器的 UDFs 到 FLUENT

1.　DEFINE_ADJUST

使用 DEFINE_ADJUST 宏定义的 UDF，当被通译或编译之后，你在 DEFINE 宏的自变量中规定的函数名将在 **User-Defined Function Hooks** 面板（如图 18.3.4.1）中变地可视和可

选。

Define → User-Defined → Function Hooks…

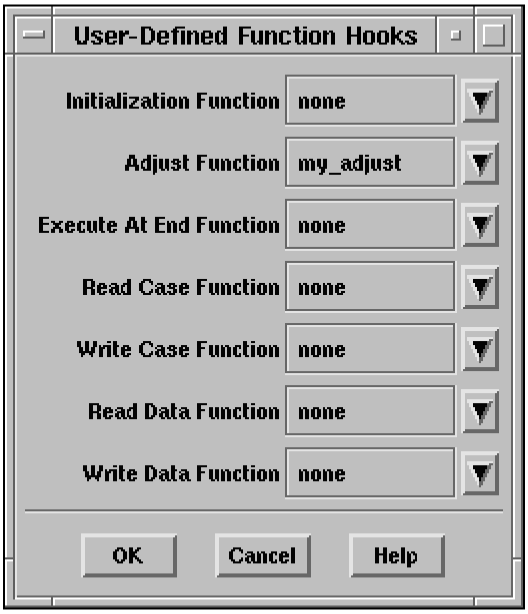


图 18.3.4.1：User-Defined Function Hooks 面板

要连接 UDF 到 FLUENT 中，在 **User-Defined Function Hooks** 面板上的 **Adust Function** 的下拉菜单中选择函数的名字（例如，my_adjust），然后点击 **OK**。

2. DEFINE_EXECUTE_AT_END

使用 DEFINE_EXECUTE_AT_END 宏定义的 UDF，当被通译或编译之后，你在 DEFINE 宏的自变量中规定的名字在 FLUENT 的 **User-Defined Function Hooks** 面板中将是可视与可选的。

Define → User-Defined → Function Hooks…

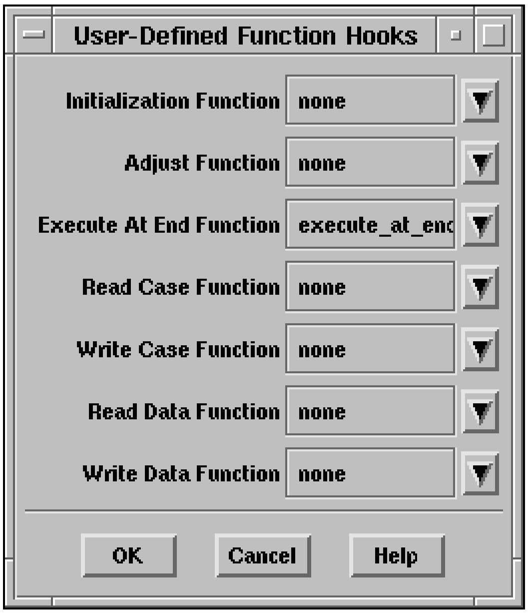


图 18.3.4.2：User-Defined Function Hooks 面板

要连接 UDF 到 FLUENT，在 **User-Defined Function Hooks** 面板上的 **Execut At End Function** 的下拉菜单中选择函数的名字（例如，execute_at_end），然后点击 **OK**。

3. DEFINE_INIT

一旦通译或编译了 DEFINE_INIT UDF，你在宏的自变量中规定的函数的名字在 FLUENT 的 **User-Defined Function Hooks** 面板中将是可视与可选的。

Define → User-Defined → Function Hooks…

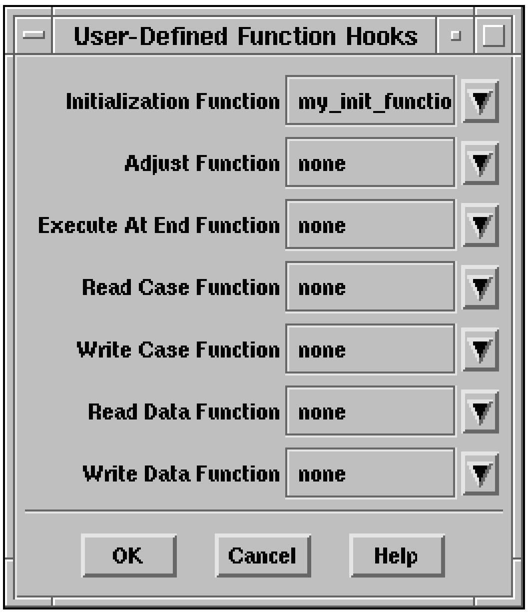


图 18.3.4.3：User-Defined Function Hooks 面板

要连接 UDF 到 FLUENT，在 **User-Defined Function Hooks** 面板上的 **Initialization Function** 的下拉菜单中选择函数的名字（例如，my_init_function），然后点击 **OK**。

4. DEFINE_ON_DEMAND

使用 DEFINE_ON_DEMAND 定义的 UDF，当被通译或编译之后，你在 DEFINE 宏的自变量中规定的名字将在 FLUENT 的 **User-Defined Function Hooks** 面板中变地可视与可选。

Define → User-Defined → Execute On Demand…



图 18.3.4.4：Execute On Demand 面板

要连接 UDF 到 FLUENT，在 **Execute On Demand** 面板上的 **Function** 的下拉菜单中选择函数的名字（例如，update），然后点击 **Execute**，此时 FLUENT 将立即执行 UDF，最后点击 **Close** 关掉面板。

5. DEFINE_RW_FILE

使用 DEFINE_RW_FILE 定义的 UDF，当被通译或编译之后，你在 DEFINE 宏的自变量中规定的函数名将在 FLUENT 的 **User-Defined Function Hooks** 面板中变地可视与可选。

Define → User-Defined → Function Hooks…

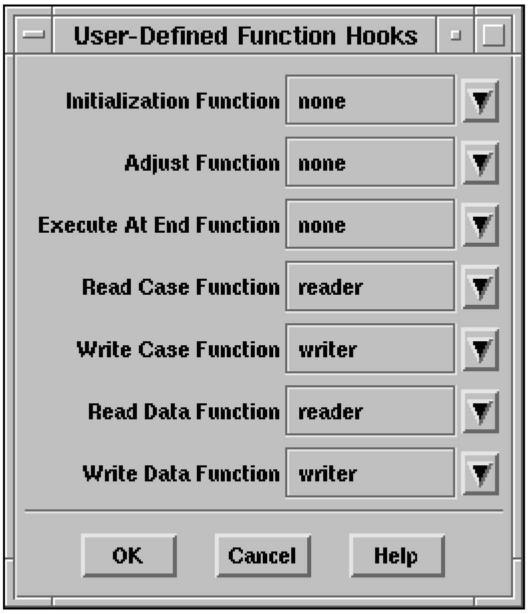


图 18.3.4.5：User-Defined Function Hooks 面板

要连接 UDF 到 FLUENT，在 **User-Defined Function Hooks** 面板上的某个合适的函数（例如，**Read Case Function**）的下拉菜单中选择函数的名字（例如，reader），然后点击 **OK**。下面给出每个 RW File 函数选项的解释。

- 当你把一个 case 文件读到 FLUENT 中时，**Read Case Function** 将被访问。它将把从 case 文件中读进的用户化函数列入清单。
- 当你从 FLUENT 中写一个 case 文件时，**Write Case Function** 将被访问。它将把写入 case 文件的用户化函数列入清单。
- 当你把一个 data 文件读到 FLUENT 中时，**Read Data Function** 将被访问。它将把从 data 文件中读进的用户化函数列入清单。
- 当你从 FLUENT 中写一个 data 文件时，**Write Data Function** 将被访问。它将把写入 data 文件的用户化函数列入清单。

6. 用户定义内存的存储

你可以把由你的 UDF 计算所得的值存储到内存里，以便以后可以获得这些值，或者通过一个 UDF，或者在 FLUENT 的后处理时都可以再次得到它们。为了能够利用这个内存，你将需要在 **User-Defined Memory**（用户定义的内存）面板中指定 **Number of User-Defined Memory Locations**（用户定义内存位置数）的值。

Define → User-Defined → Memory…

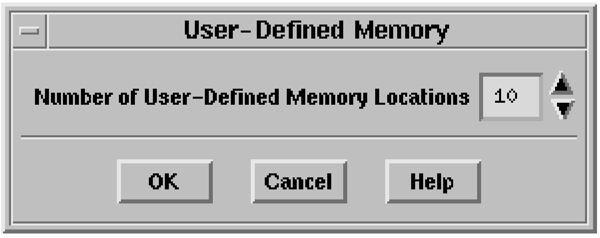


图 18.3.4.6：User-Defined Memory 面板

宏 C_UDMI 和 F_UDMI 分别可以用在你的 UDF 中去访问一个位于某个单元或面内的特殊用户定义内存的位置。

框中的值储存在用户定义的内存中，若写 data 文件，择这些值将被保存到这个文件里。这些框也会出现在 FLUENT 的后处理面板中的下拉菜单的 **User Defined Memory…**类里。基于内存位置的指数，它们将被命名为 udm-0，udm-1 等等。内存位置的总数不能超过 500。用户定义内存位置的数越大，对系统内存的要求就越高。

18.3.5 连接物种模型的 UDFs 到 FLUENT

1. DEFINE_CHEM_STEP

使用 DEFINE_CHEM_STEP 宏定义的 UDF，在被通译或编译后，你在 DEFINE 宏的自变量中规定的名字将会在 FLUENT 的 User-Defined Function Hooks 面板中变地可视与可选。

Define → User-Defined → Function Hooks…

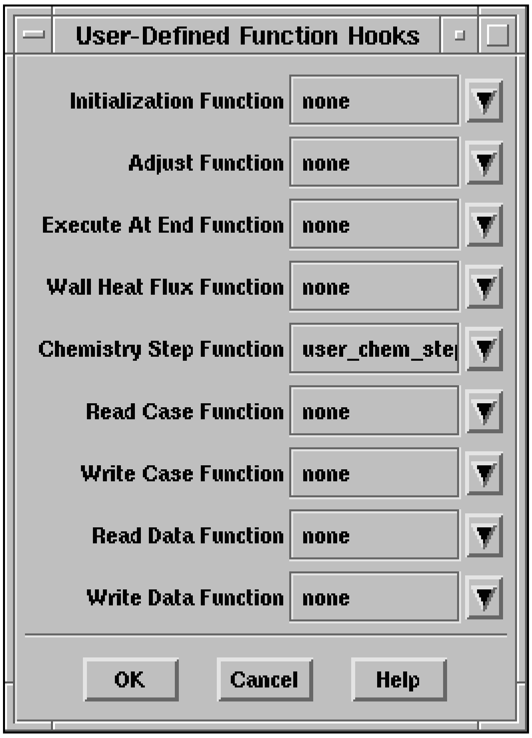


图 18.3.5.1：User-Defined Function Hooks 面板

！EDC 和 PDF 输运模型必须能够连接化学步的 UDFs。

要连接 UDF 到 FLUENT 中，在 **User-Defined Function Hooks** 面板上的 **Chemistry Step Function** 的下拉菜单中选择函数的名字（例如，user_chem._step），然后点击 **OK**。

2. DEFINE_DELTAT

使用 DEFINE_DELTAT 宏定义的 UDF，在被通译或编译后，你在 DEFINE 宏的自变量中规定的函数名将会在 FLUENT 的 **Iterate** 面板中变地可视与可选。

Define → Iterate…



图 18.3.5.2：Iterate 面板

要连接 UDF 到 FLUENT 中，首先你将需要在 Iterate 面板上选择 Time Stepping Method 下的 **Adaptive** 项，然后在 **AdaptiveTime Stepping** 下面的 **User-Defined Time Step** 的下拉菜单中选择函数的名字（例如，mydeltat）。

3. DEFINE_DIFFUSIVITY

使用 DEFINE_DIFFUSIVITY 宏定义的 UDF，在被通译或编译后，你在 DEFINE 宏的自变量中规定的函数名将会在 FLUENT 的 **Materials** 面板中变地可视与可选（如图 18.4.5.3 所示）。要连接 UDF 到 FLUENT，首先在 **Materials** 面板上的 **Properties** 一栏中的 **Mass Diffusivity** 的下拉菜单中，通过选择 **user-defined** 打开 **User-Defined Functions** 面板。

Define → Materals…

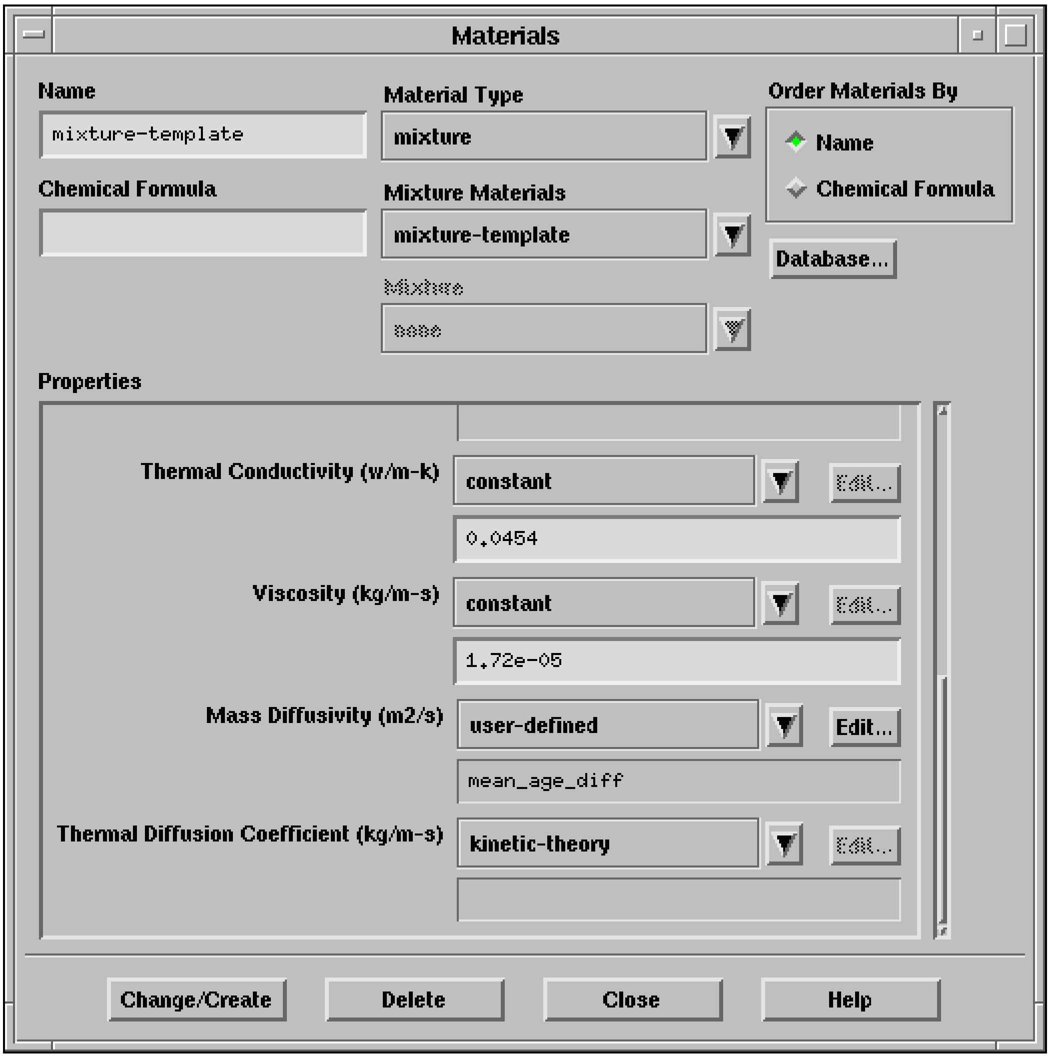


图 18.3.5.3：Materials 面板

下一步，在 **User-Defined Functions** 面板上（如图 18.3.5.4 所示），从 UDFs 列中选择函数名（例如，mean_age_diff）。



图 18.3.5.4：User-Defined Functions 面板

！EDC 和 PDF 输运模型必须能够连接化学步的 UDFs。

要连接 UDF 到 FLUENT 中，在 User-Defined Function Hooks 面板上的 Chemistry Step **Function** 的下拉菜单中选择函数的名字（例如，user_chem._step），然后点击 **OK**。

4. DEFINE_DOM_DIFFUSE_REFLECTIVITY

使用 DEFINE_DOM_DIFFUSE_REFLECTIVITY 宏定义的 UDF，在被通译或编译后，你在 DEFINE 宏的自变量中规定的函数名将会在 FLUENT 的 **User-Defined Function Hooks** 面板中变地可视与可选（如图 18.3.5.5 所示）。

Define → User-Defined → Function Hooks…

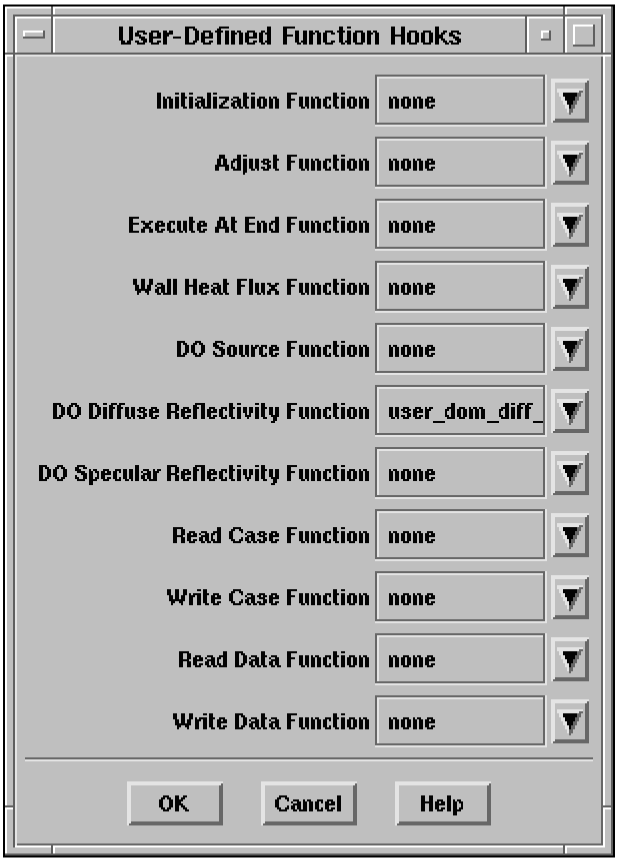


图 18.3.5.5：User-Defined Function Hooks 面板

！离散纵坐标（Descrete Ordinates）辐射模型必须在 **Radiation Model** 面板中被激活。

要连接 UDF 到 FLUENT 中，在 **User-Defined Function Hooks** 面板上的 **DO Diffuse Reflectivity Function** 的下拉菜单中选择函数的名字（例如，user_dom_diff_refl），然后点击 **OK**。

5. DEFINE_DOM_SOURCE

使用 DEFINE_DOM_SOURCE 宏定义的 UDF，在被通译或编译后，你在 DEFINE 宏的自变量中规定的函数名将会在 FLUENT 的 **User-Defined Function Hooks** 面板中变地可视与可选（如图 18.3.5.6 所示）。

！离散纵坐标辐射模型必须被激活。

Define → User-Defined → Function Hooks…

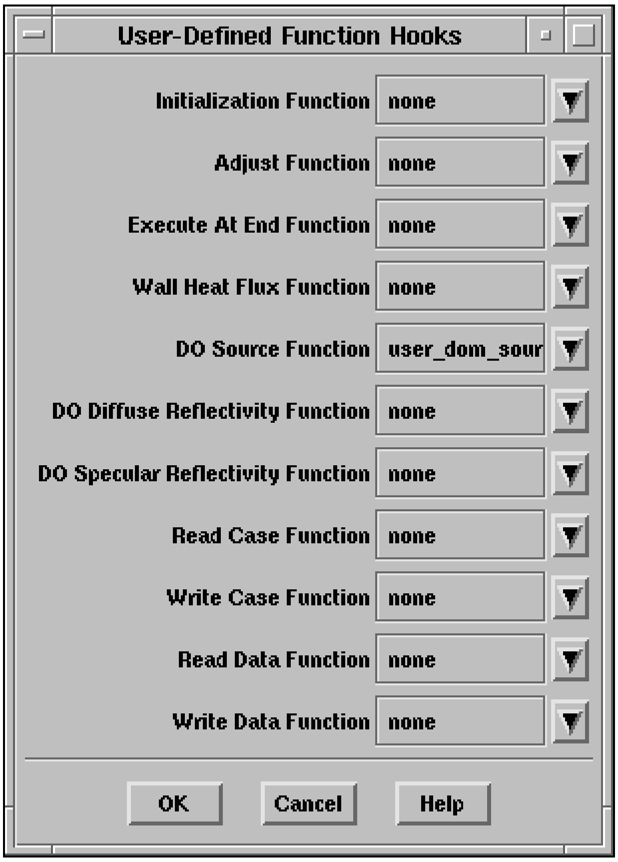


图 18.3.5.6：User-Defined Function Hooks 面板

要连接 UDF 到 FLUENT 中，在 **User-Defined Function Hooks** 面板上的 **DO Source Function** 的下拉菜单中选择函数的名字（例如，user_dom_source），然后点击 **OK**。

6. DEFINE_DOM_SPECULAR_REFLECTIVITY

使用 DEFINE_DOM_SPECULAR_REFLECTIVITY 宏定义的 UDF，在被通译或编译后，你在 DEFINE 宏的自变量中规定的函数名将会在 FLUENT 的 **User-Defined Function Hooks** 面板中变地可视与可选（如图 18.3.5.7 所示）。

Define → User-Defined → Function Hooks…

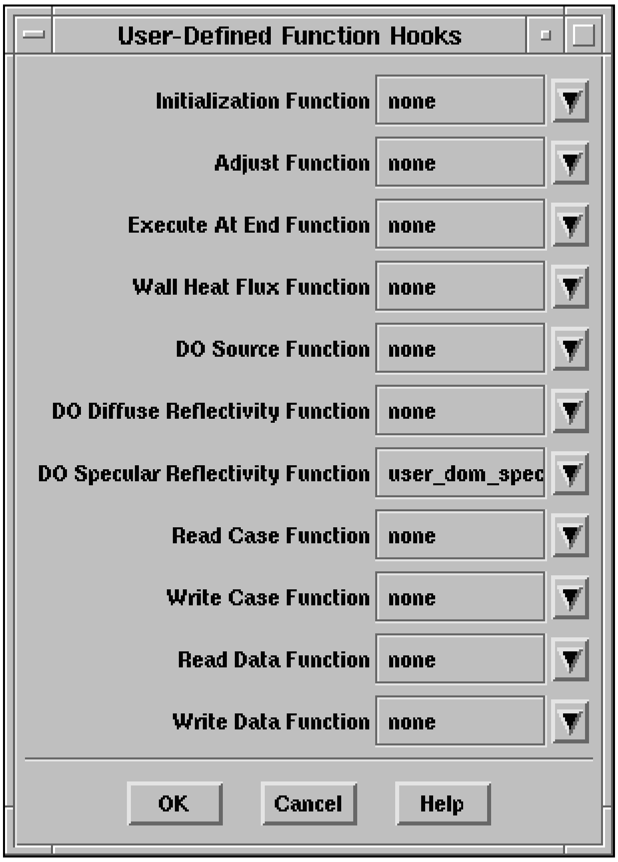


图 18.3.5.7：User-Defined Function Hooks 面板

！离散纵坐标辐射模型必须在 **Radiation Model** 面板中被激活。

要连接 UDF 到 FLUENT 中，在 **User-Defined Function Hooks** 面板上的 **DO Specular Reflectivity Function** 的下拉菜单中选择函数的名字（例如，user_dom_spec_refl），然后点击 **OK**。

7. DEFINE_HEAT_FLUX

使用 DEFINE_HEAT_FLUX 宏定义的 UDF，在被通译或编译后，你在 DEFINE 宏的自变量中规定的函数名将会在 FLUENT 的 **User-Defined Function Hooks** 面板中变地可视与可选（如图 18.3.5.8 所示）。

Define → User-Defined → Function Hooks…

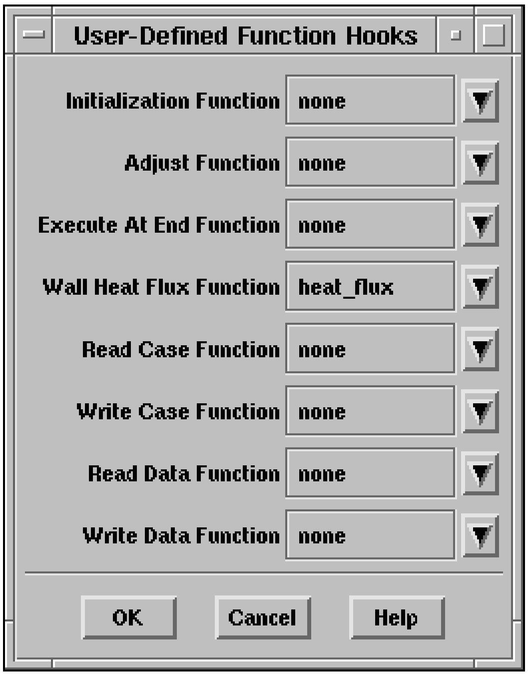


图 18.3.5.8：User-Defined Function Hooks 面板

！能量方程（Energy Equation）必须是激活的。

要连接 UDF 到 FLUENT 中，在 **User-Defined Function Hooks** 面板上的 **Wall Heat Flux Function** 的下拉菜单中选择函数的名字（例如，heat_flux），然后点击 **OK**。

8. DEFINE_NET_REACTION_RATE

使用 DEFINE_NET_REACTION_RATE 宏定义的 UDF，在被通译或编译后，你在 DEFINE 宏的自变量中规定的函数名将会在 FLUENT 的 **User-Defined Function Hooks** 面板中变地可视与可选（如图 18.3.5.9 所示）。

Define → User-Defined → Function Hooks…

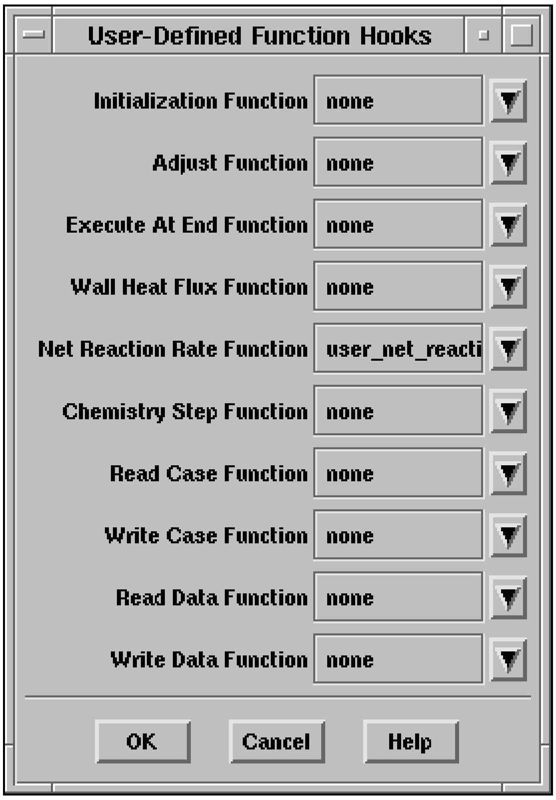


图 18.3.5.9：User-Defined Function Hooks 面板

！净反应率的 UDFs 既可以用于表面化学模型中，也可用于 EDC 和 PDF 输运模型中。要激活 PDF 输运模型，请在 **Species Model** 面板上的选择 **Composition PDF Transport** 和 **Volumetric** 的反应。要激活 EDC 模型，请在 **Species Model** 面板上的选择 **Species Transport** 和 **Volumetric** 的反应，以及选择 **Turbulence-Chemistry Interaction** 下的 EDC。

要连接 UDF 到 FLUENT 中，在 **User-Defined Function Hooks** 面板上的 **Net Reaction Rate Function** 的下拉菜单中选择函数的名字（例如，user_net_reaction_rate），然后点击 **OK**。

9. DEFINE_NOX_RATE

使用 DEFINE_NOX_RATE 宏定义的 UDF，在被通译或编译后，你在 DEFINE 宏的自变量中规定的函数名将会在 FLUENT 的 **NOx Model** 面板中变地可视与可选（如图 18.3.5.10 所示）。

Define → Models → Pollutants → NOx…

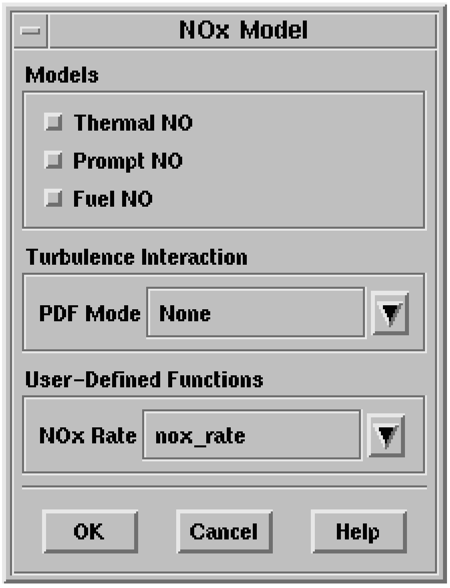


图 18.3.5.10：NOx Model 面板

要连接 UDF 到 FLUENT 中，在 **NOx Model** 面板上的 User-Defined Function 下面的 **NOX Rate** 的下拉菜单中选择函数的名字（例如，nox_rate），然后点击 **OK**。

10. DEFINE_PR_RATE

使用 DEFINE_PR_RATE 宏定义的 UDF，在被通译或编译后，你在 DEFINE 宏的自变量中规定的函数名将会在 FLUENT 的 **User-Defined Function Hooks** 面板中变地可视与可选（如图 18.3.5.11 所示）。

Define → User-Defined → Function Hooks…

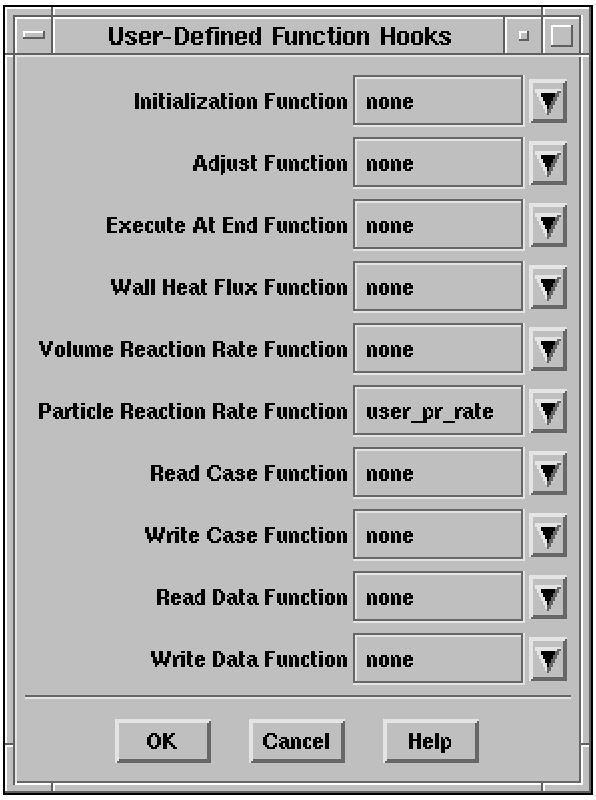


图 18.3.5.11：User-Defined Function Hooks 面板

！在 **Species Model** 面板上的 **Reactions** 下面选择 **Volumetric** 和 **Particle Surface**，这样你可以连接你的 UDF，但在做这步之前你必须激活粒子表面反应的选项。

要连接 UDF 到 FLUENT 中，在 **User-Defined Function Hooks** 面板上的 **Particle Reaction Rate Function** 的下拉菜单中选择函数名（例如，user_pr_rate），然后点击 **OK**。

11. DEFINE_PRANDTL

使用 DEFINE_PRANDTL 宏定义的 UDF，在被通译或编译后，你在 DEFINE 宏的自变量中规定的函数名将会在 FLUENT 的 **Viscous Model** 面板中变地可视与可选（如图 18.3.5.12 所示）。

Define → Models → Viscous…

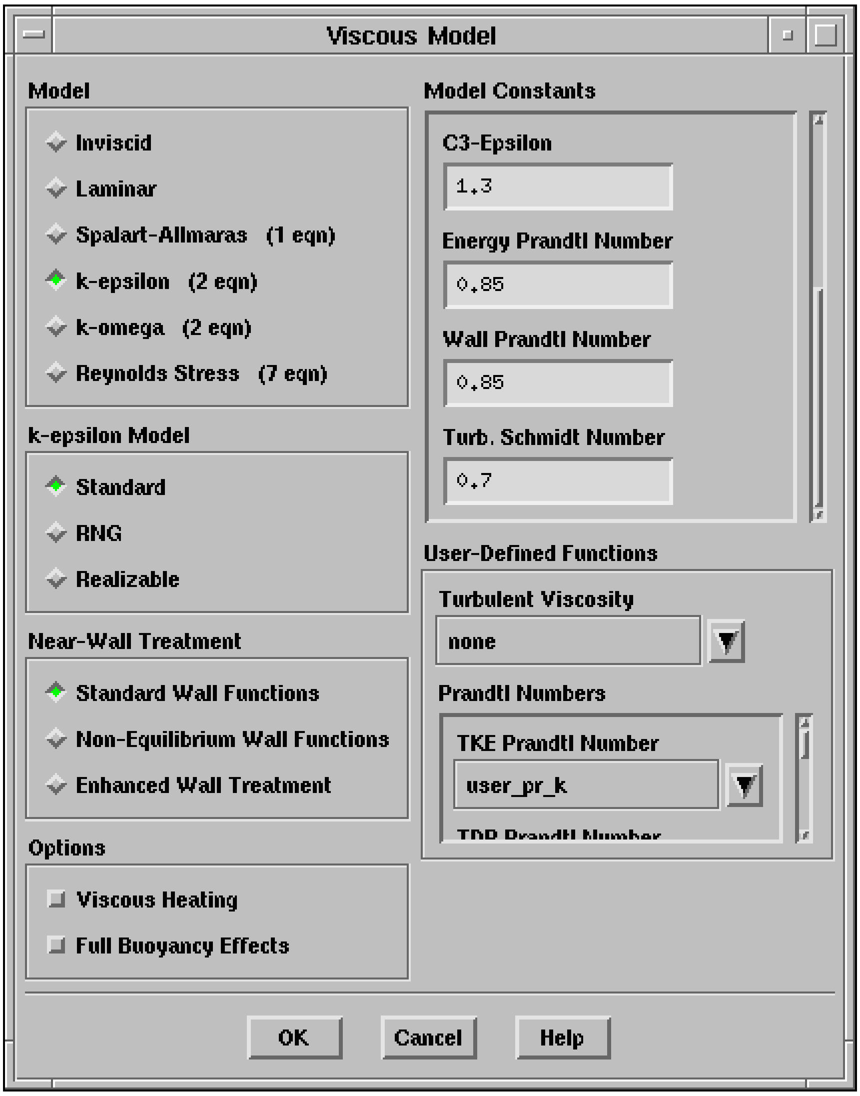


图 18.3.5.12：Viscous 面板

要连接 UDF 到 FLUENT 中，在 **Viscous Model** 面板上的 **User-Defined Function** 下面的 **TKE Prandtl Number** 的下拉菜单中选择函数名（例如，user_pr_k），然后点击 **OK**。

12. DEFINE_PROFILE

使用 DEFINE_PROFILE 宏定义的 UDF，在被通译或编译后，你在 DEFINE 宏的自变量中规定的函数名将会在 FLUENT 的相应的边界条件面板中变地可视与可选。

例如，如果你的 UDF 定义了一个速度进口边界条件，那么为了连接这个函数到 FLUENT 中，在 **Velocity Inlet** 面板（如图 18.3.5.13）上的相关下拉菜单中选择函数名（例如，inlet_x_velocity），然后点击 **OK**。请注意，在下拉菜单中，单词 **udf** 排在 UDF 的名字前面（例如，udf inlet_x_velocity）。

Define → Boundary Conditions…

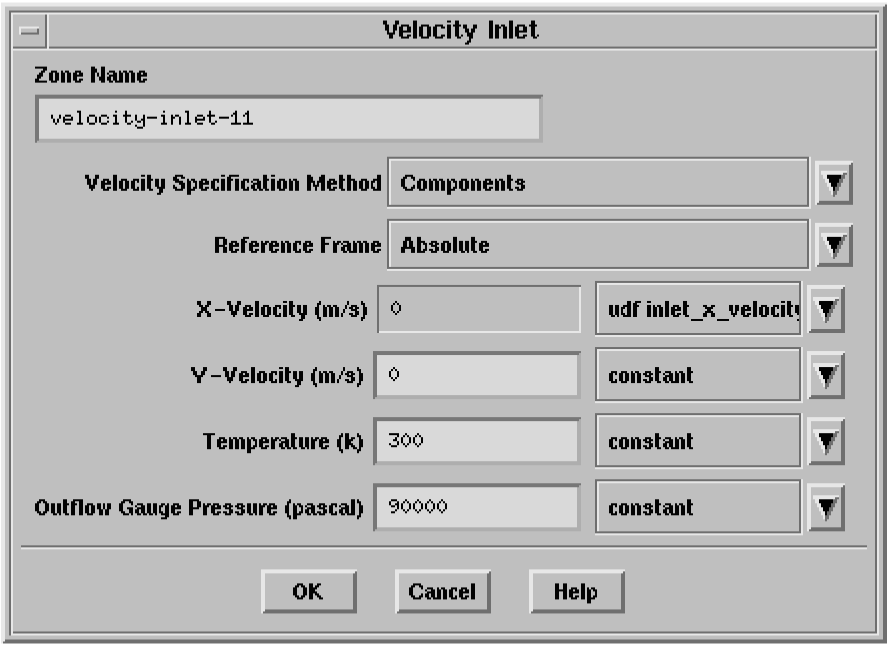


图 18.3.5.13：Velocity Inlet 面板

如果你使用你的 UDF 在一个单元域内规定一个固定值，那么你将需要打开 **Fixed Values** 选项，并在 **Fluid** 或 **Solid** 面板上的相关下拉菜单中选择你的函数名。

13.　　DEFINE_PROPERTY

使用 DEFINE_PROPERTY 宏定义的 UDF 是关于材料特性的函数，UDF 在被通译或编译后，函数名将会在 FLUENT 的 **User-Defined Functions** 面板（如图 18.3.5.15）中变地可视与可选。要连接 UDF 到 FLUENT，首先在 **Materials** 面板（如图 18.3.5.14）上的 **Properties** 一栏中的适当的特性，（如，**Viscosity**） 的下拉菜单中，通过选择 **user-defined** 打开 **User-Defined Functions** 面板。

Define → Materals…

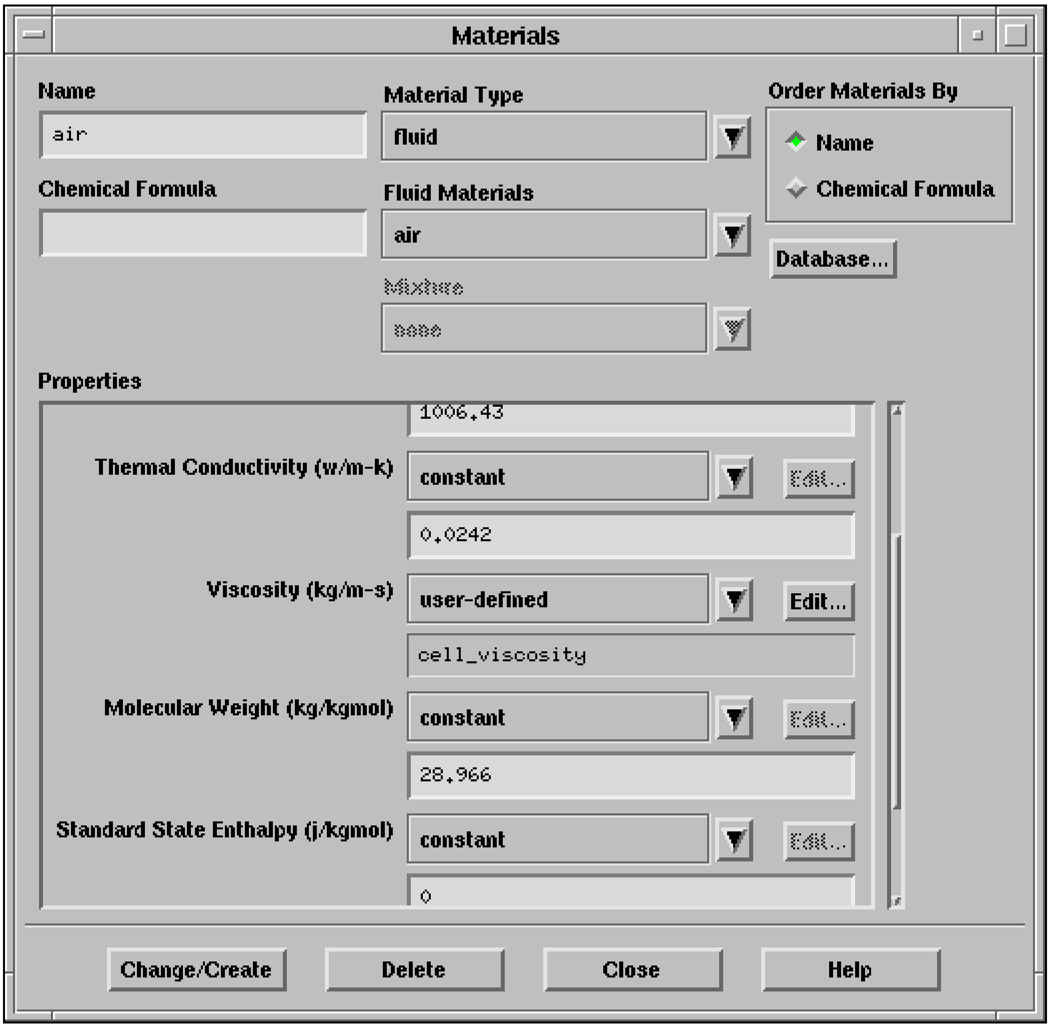


图 18.3.5.14：Materials 面板

下一步，在 **User-Defined Functions** 面板上，从 UDFs 列中选择函数名（例如，cell_viscosity），然后点击 **OK**。之后，这个函数名会在 **Materials** 面板中你所选的特性（如，Viscosity）的下面出现。



图 18.3.5.15：User-Defined Functions 面板

！如果你打算使用一个 UDF 定义密度，那么请注意，随着密度变量的增大，解的收敛将会变弱。考虑到密度是一个与压力有关的函数及它在空间上是变化的，规定一个压力法则或多相行为可能导致解发散。建议限制关于密度的 UDFs 的使用范围，对于密度变化比较小的弱压缩流动可以使用。

14. DEFINE_SCAT_PHASE_FUNC

一旦通译或编译了 DEFINE_SCAT_PHASE_FUNC UDF，函数名将会在 FLUENT 的 **User-Defined Functions** 面板（如图 18.3.5.17）中变地可视与可选。要连接 UDF 到 FLUENT，首先你要在从 **Materials** 面板（如图 18.3.5.16）中打开 **User-Defined Functions** 面板，方法是在 **Properties** 一栏中 **Scattering Phase Function** 的下拉菜单中选择 **user-defined**。

Define → Materals…

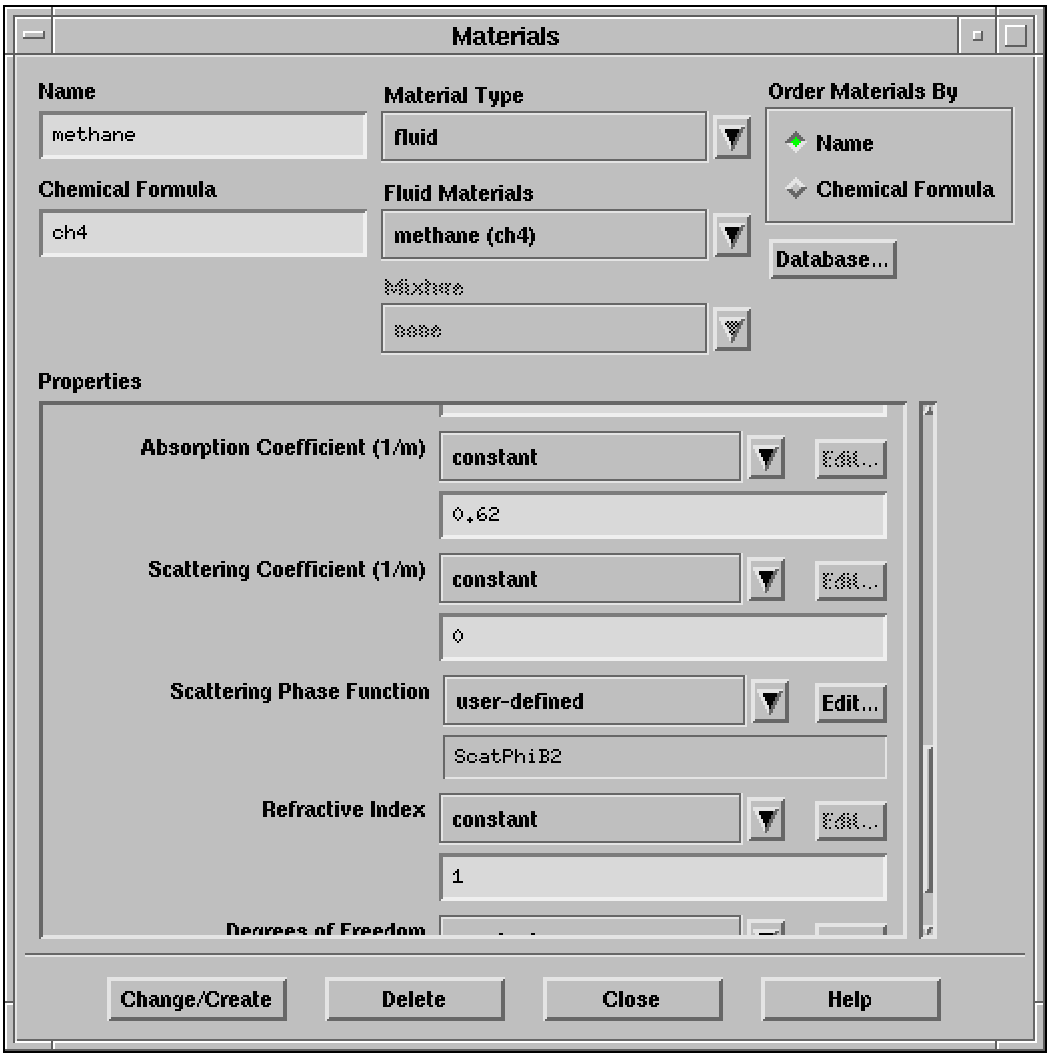


图 18.3.5.16：Materials 面板

！离散纵坐标的辐射模型必须从 **Radiation Model** 面板中被激活。

下一步，在 **User-Defined Functions** 面板上，从 UDFs 列中选择函数名（例如，ScatPhiB2），

然后点击 **OK**。之后，这个函数名会显示在 **Materials** 面板上的 **Scattering Phase Function** 特性的下面。

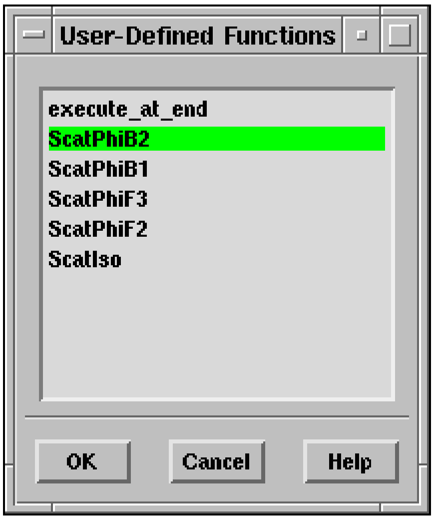


图 18.3.5.17：User-Defined Functions 面板

！如果你打算使用一个 UDF 定义密度，那么请注意，随着密度变量的增大，解的收敛将会变弱。考虑到密度是一个与压力有关的函数及它在空间上是变化的，规定一个压力法则或多相行为可能导致解发散。建议限制关于密度的 UDFs 的使用范围，对于密度变化比较小的弱压缩流动可以使用。

15. DEFINE_SOURCE

一旦通译或编译了 DEFINE_SOURCE UDF，函数名将会在 FLUENT 的 **Fluid** 或 **Solid** 面板中变地可视与可选。要连接 UDF 到 FLUENT，首先打开 **Fluid** 或 **Solid** 面板（如图 18.3.5.18）中的 **Source Terms** 项。

Define → Boundary Conditions…

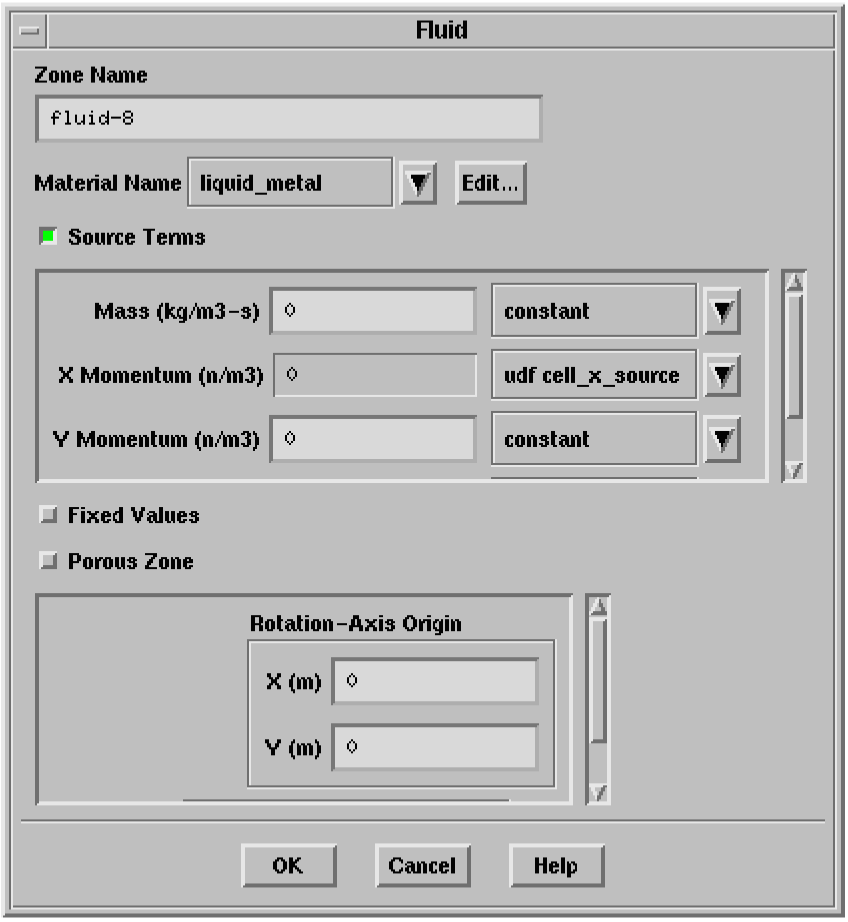


图 18.3.5.18：Fluid 面板

下一步，在 **Fluid** 或 **Solid** 面板上的适当下拉菜单（如，**X Momentum**）里选择函数的名字（例如，cell_x_source），然后点击 **OK**。请注意，在下拉菜单中，单词 **udf** 出现于 UDF 名字的前面（如，udf cell_x_source）。

16. DEFINE_SR_RATE

使用 DEFINE_HEAT_FLUX 宏定义的 UDF，一旦被通译或编译，函数名将会在 FLUENT 的 **User-Defined Function Hooks** 面板（如图 18.3.5.19 所示）中变地可视与可选。

Define → User-Defined → Function Hooks…

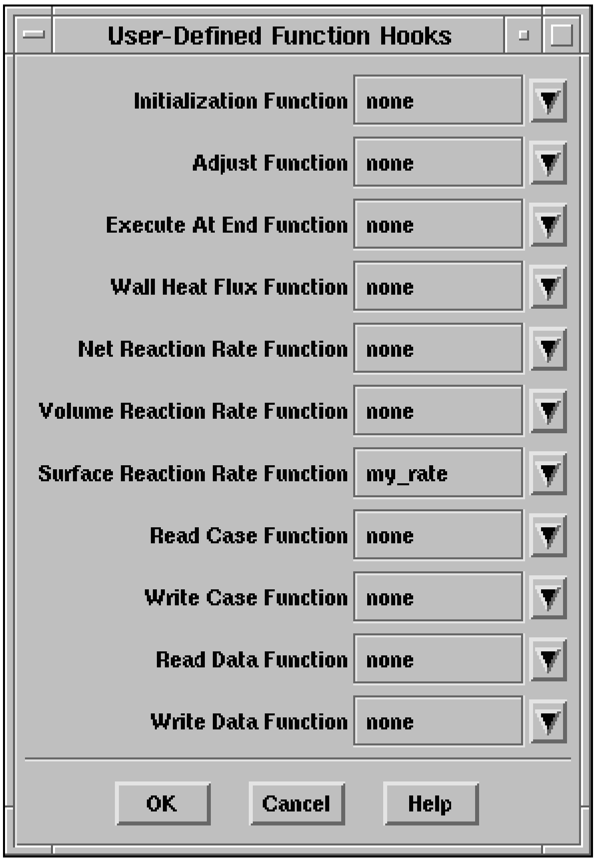


图 18.3.5.19：User-Defined Function Hooks 面板

！在 **Species Model** 面板上，选择 **Reactions** 下面的 **Volumetric** 和 **Wall Surface** 可以连接 UDF，在这之前，必须激活壁面表面反应选项。

要连接 UDF 到 FLUENT 中，在 **User-Defined Function Hooks** 面板上的 **Surface Reaction Rate Function** 的下拉菜单中选择函数的名字（例如，my_rate），然后点击 **OK**。

17. DEFINE_TURB_PREMIX_SOURCE

一旦通译或编译了 DEFINE_TURB_PREMIX_SOURCE UDF，函数名将会在 FLUENT 的 **User-Defined Function Hooks** 面板（如图 18.3.5.20 所示）中变地可视与可选。

Define → User-Defined → Function Hooks…

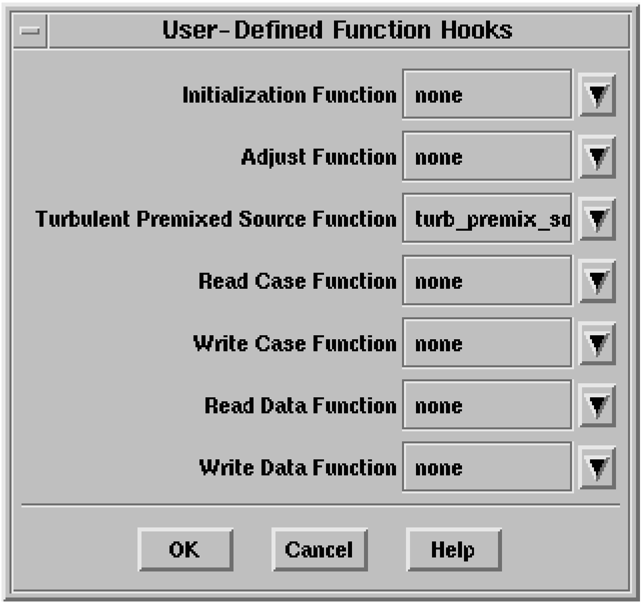


图 18.3.5.20：User-Defined Function Hooks 面板

！ 一个预混合材料燃烧的模型必须在 **Species Model** 面板中被激活。

要连接 UDF 到 FLUENT 中，在 **User-Defined Function Hooks** 面板上的 **Turbulent Premixed Source Function** 的下拉菜单中选择函数的名字（例如，turb_premix_source），然后点击 **OK**。

18. DEFINE_TURBULENT_VISCOSITY

一旦通译或编译了 DEFINE_TURBULENT_VISCSITY UDF，函数名将会在 FLUENT 的 **Viscous Model** 面板（如图 18.3.5.21 所示）中变地可视与可选。

Define → Models → Viscous…

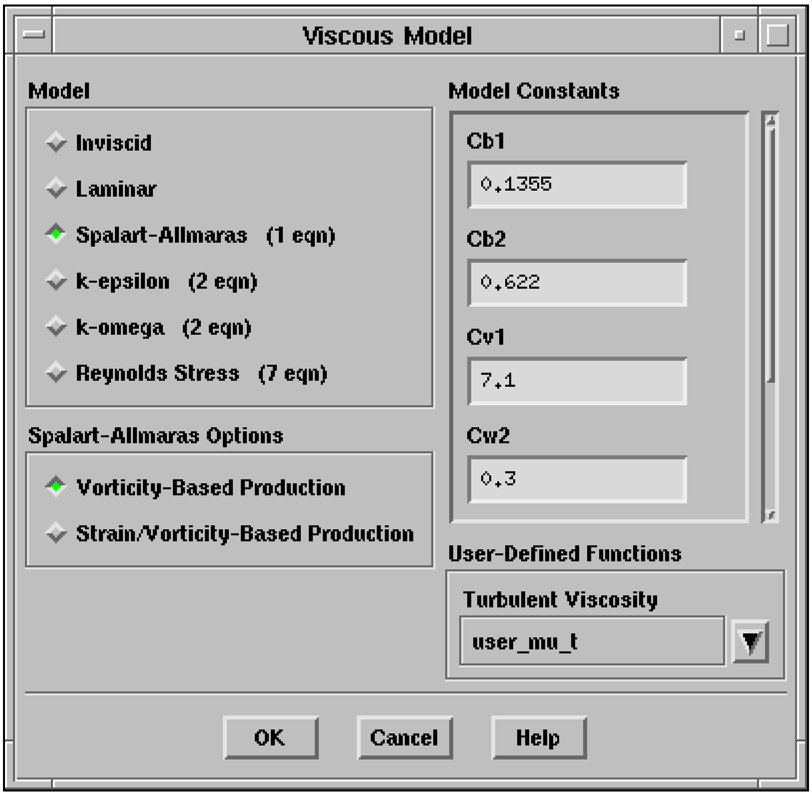


图 18.3.5.21：Viscous 面板

要连接 UDF 到 FLUENT 中，在 **Viscous Model** 面板上的 **User-Defined Function** 下面的 **Turbulence Viscosity** 的下拉菜单中选择函数名（例如，user_mu_t），然后点击 **OK**。

19. DEFINE_UDS_FLUX

一旦通译或编译了 DEFINE_UDS_FLUX UDF，函数名将会在 FLUENT 的 **User-Defined Scalars** 面板（如图 18.3.5.22 所示）中变地可视与可选。

Define → User-Defined → Scalars…

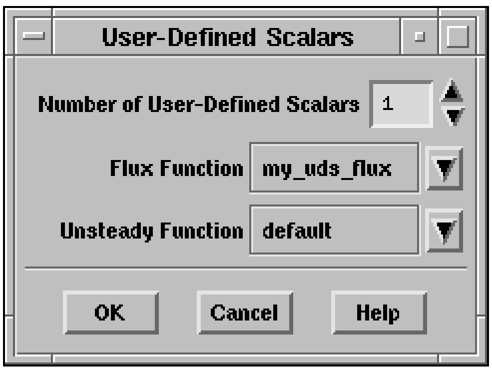


图 18.3.5.22：User-Defined Scalars 面板

要连接 UDF 到 FLUENT 中，首先在 **User-Defined Scalars** 面板中指定 **Number of User-Defined Scalars**（用户定义标量的数）的值。随着用户定义标量的数的输入，面板将显示一个 **Flux Function**（通量函数）的设置。其次，在 **Flux Function** 的下拉菜单里选择函数

名（例如，my_uds_flux），然后点击 **OK**。

20. DEFINE_UDS_UNSTEADY

一旦通译或编译了 DEFINE_UDS_UNSTEADY UDF，函数名将会在 FLUENT 的 **User-Defined Scalars** 面板（如图 18.3.5.23）中变地可视与可选。

Define → User-Defined → Scalars…

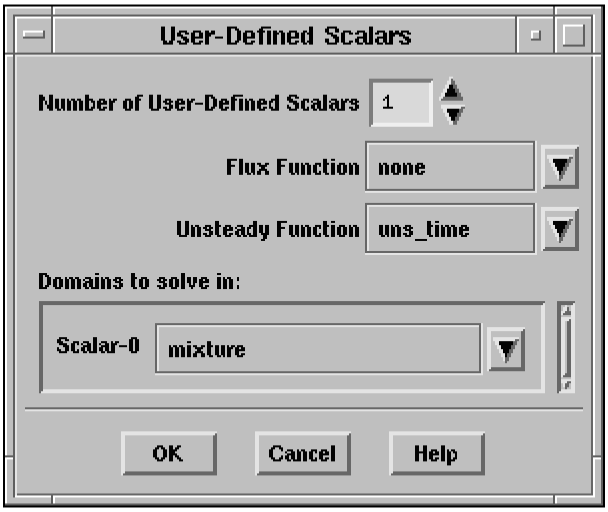


图 18.4.5.23：User-Defined Scalars 面板

要连接 UDF 到 FLUENT 中，首先在 **User-Defined Scalars** 面板中指定 **Number of User-Defined Scalars** 的值。随着用户定义标量的数的输入，面板将显示一个 **Unsteady Function**（非稳定函数）的设置。其次，在 **Unsteady Function** 的下拉菜单里选择函数名（例如，uns_time），然后点击 **OK**。

21. DEFINE_VR_RATE

一旦通译或编译了 DEFINE_VR_RATE UDF，函数名将会在 FLUENT 的 **User-Defined Function Hooks** 面板（如图 18.3.5.24 所示）中变地可视与可选。

Define → User-Defined → Function Hooks…

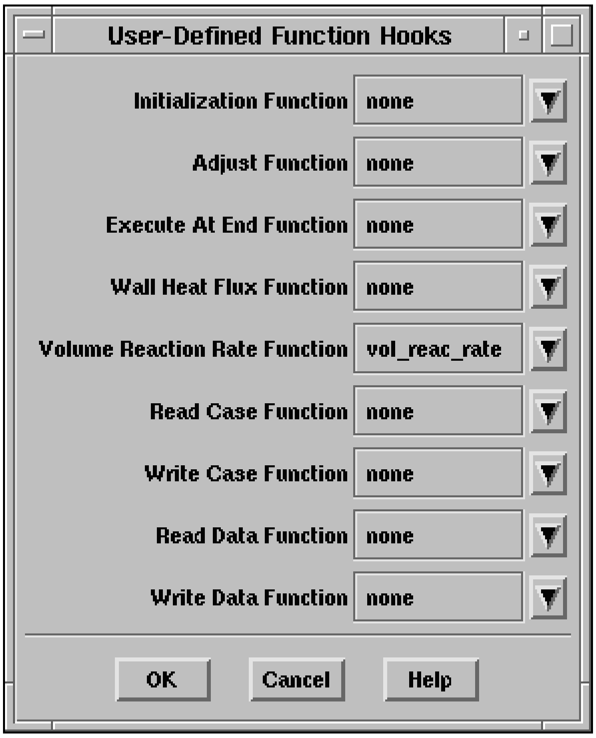


图 18.3.5.24：User-Defined Function Hooks 面板

! 在 **Species Model** 面板中的 Reactions 下选择 **Volumetric** 可以连接 UDF，但在这之前，必须先打开体积反应选项。

要连接 UDF 到 FLUENT 中，在 **User-Defined Function Hooks** 面板上的 **Volume Reaction Rate Function** 的下拉菜单中选择函数的名字（例如，vol_reac_rate），然后点击 **OK**。

18.3.6 连接多相的 UDFs 到 FLUENT

1. DEFINE_CAVITATION_RATE

一旦通译或编译了 DEFINE_CAVITATION_RATE UDF，函数的名字将会在 FLUENT 的 **User-Defined Function Hooks** 面板（如图 18.3.6.1 所示）中变地可视与可选。请注意，气蚀率的 UDFs 仅适用于混合物的多相模型。

要连接 UDF 到 FLUENT 中，首先，在 **Multiphase Model** 面板（如图 18.3.6.2）中激活 **Mixture**（混合物）模型，然后在 **Interphase Mass Transfer** 下面选择 **Cavitation**。

Define → Models → Multiphase…

下一步，打开 **User-Defined Function Hooks** 面板，

Define → User-Defined → Function Hooks…

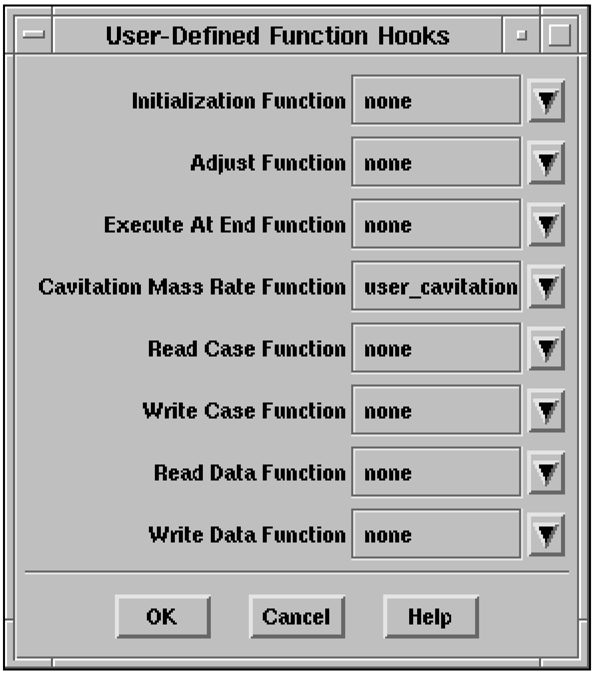


图 18.3.6.1：User-Defined Function Hooks 面板

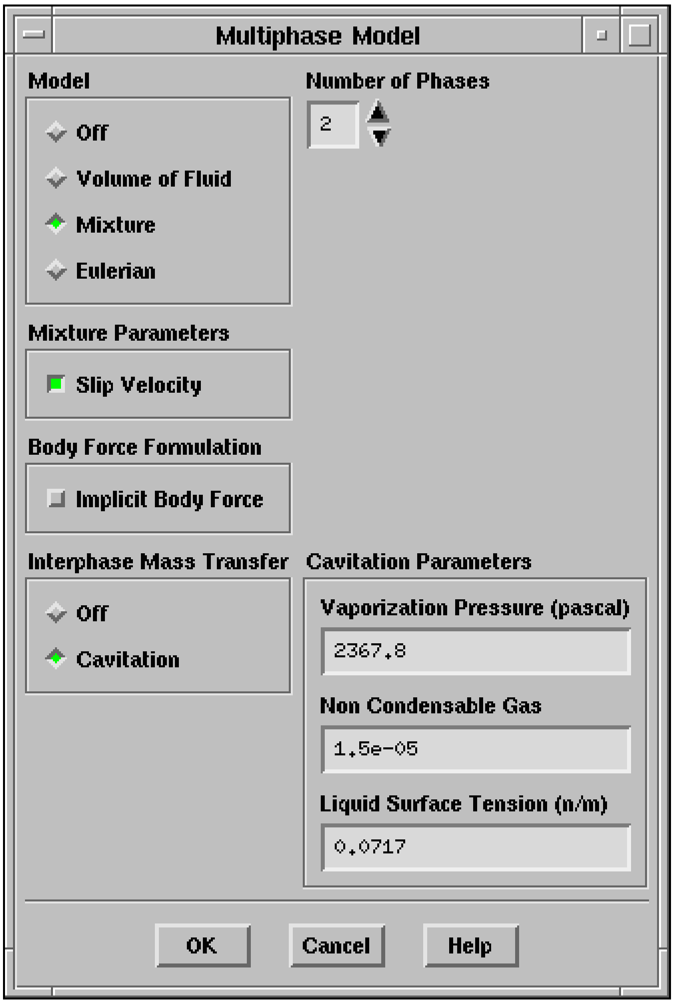


图 18.3.6.2：Multiphase Model 面板

在 **Cavitatin Mass Rate Function** 的下拉菜单中选择函数的名字（例如，user_cavitation_rate），然后点击 **OK**。

2. DEFINE_EXCHANGE_PROPERTY

一旦通译或编译了 DEFINE_EXCHANGE_PROPERTY UDF，函数的名字将会在 FLUENT 的 **User-Defined Function** 面板（如图 18.3.6.4 所示）中变地可视与可选。

用户化的质量输运 UDFs 可以用于 VOF，混合物和 Eulerian 多相模型。阻力系数 UDFs 可以用于混合物和 Eulerian 模型，而热传递和升力系数 UDFs 仅适用于 Eulerian 模型。在连接函数之前需要激活多相模型。

要连接一个交换特性的 UDF 到 FLUENT，首先需要在 **Phases** 面板中通过点击 **Interactions...**打开 **Phase Interaction** 面板（如图 18.3.6.3 所示）。

Define → Phases…

下一步，在 **Phase Interaction** 面板中点击合适的标签（如，**Drag**），然后从所需要的相应交换特性（如，**Drag Coefficient**）的下拉菜单中选择 **user-defined**。这样将打开 **User-Defined Functions** 面板。

Define → Dynamic Mesh → Parameters…

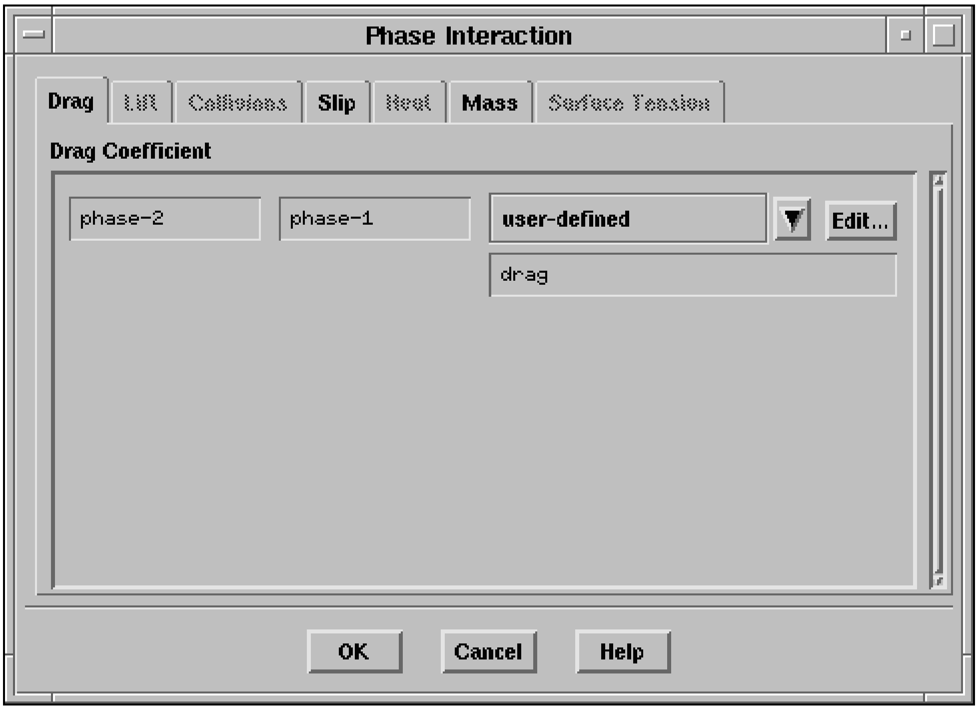


图 18.3.5.3：Phase Interaction 面板

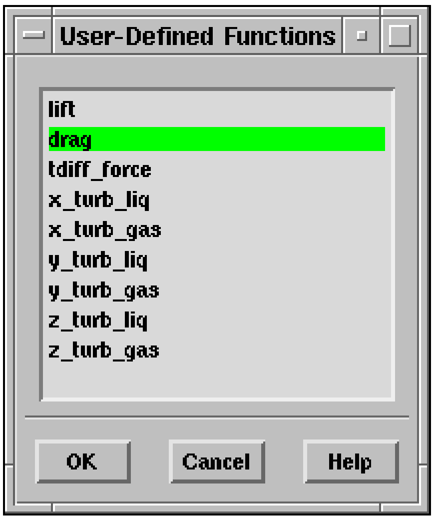


图 18.3.5.4：User-Defined Functions 面板

！为了显示关于混合物模型的阻力系数，务必在 **Multiphase Model** 面板中的 **Mixture Parameters** 下面选择 **Slip Velocity**。

最后，从显示在 **User-Defined Function** 面板中的 UDFs 列中选择函数名（如，drag），点击 **OK**。那么函数名（如，drag）将会显示于 **Phase Interaction** 面板的关于 **Drag Coefficient** 的 **user-defined** 函数的下面。

3. DEFINE_VECTOR_EXCHANGE_PROPERTY

一旦通译或编译了 DEFINE_VECTOR_EXCHANGE_PROPERTY UDF，函数名将会在 FLUENT 的 **User-Defined Functions** 面板（如图 18.3.6.6 所示）中变地可视与可选。

要连接 UDF 到 FLUENT，首先，通过点击 **Phases** 面板中 **Interactions…**打开 **Phase Interaction** 面板（如图 18.3.6.5 所示）。

Define → Phases…

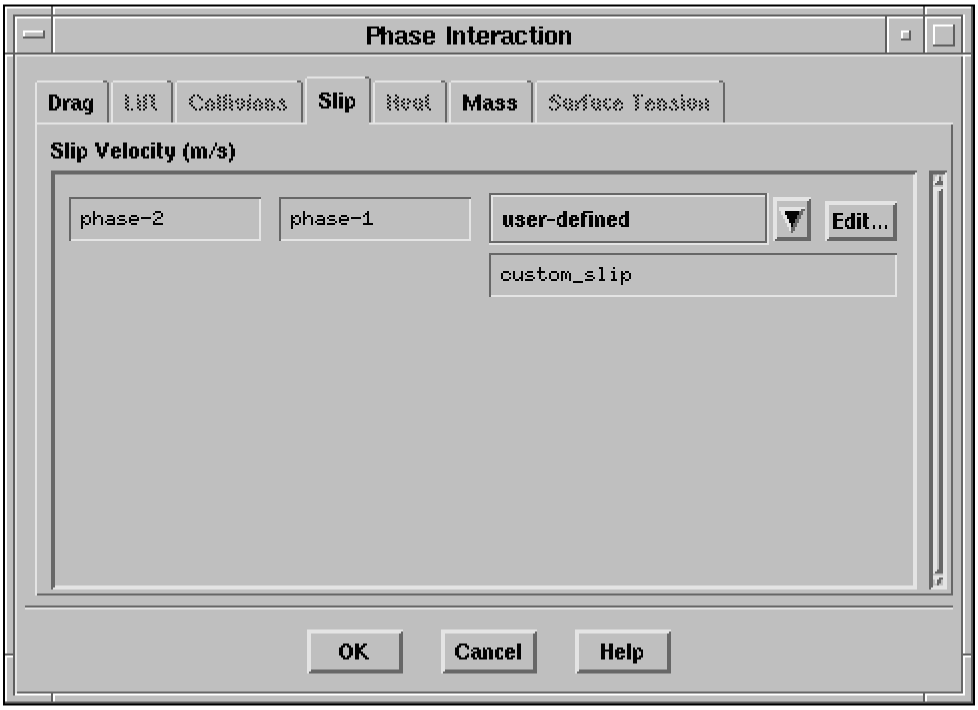


图 18.3.5.5：Phase Interaction 面板

下一步，在 **Phase Interaction** 面板中点击 **Slip** 标签，然后在关于 **Slip Velocity** 的下拉菜单中选择 **user-defined**。这样将打开 **User-Defined Functions** 面板。

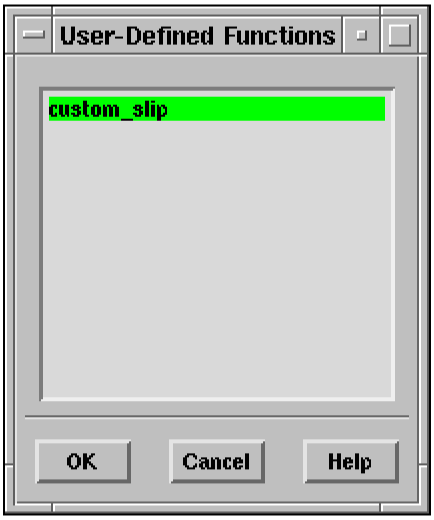


图 18.3.5.6：User-Defined Functions 面板

！滑动速度的 UDFs 仅适用于多相混合物模型。

最后，从显示在 **User-Defined Function** 面板（如图 18.3.6.4 所示）上的 UDFs 列中选择函数名（如，custom_slip），点击 **OK**。

8.4.7 连接动态网格的 UDF 到 FLUENT

1. DEFINE_CG_MOTION

一旦通译或编译了 DEFINE_CG_MOTION UDF，函数名将会在 FLUENT 的 **Dynamic Zones** 面板（如图 18.3.7.1 所示）中变地可视与可选。

要连接 UDF 到 FLUENT，首先需要激活动态网格模型。

Define → Dynamic Mesh → Parameters…

选择 **Model** 下面的 **Dynamic Mesh**，点击 **OK**。

！仅当在 **Solver** 面板中选择了 **Unsteady** 时，**Dynamic Mesh** 面板才可被访问。

其次，打开 **Dynamic Zone** 面板。

Define → Dynamic Mesh → Zones…

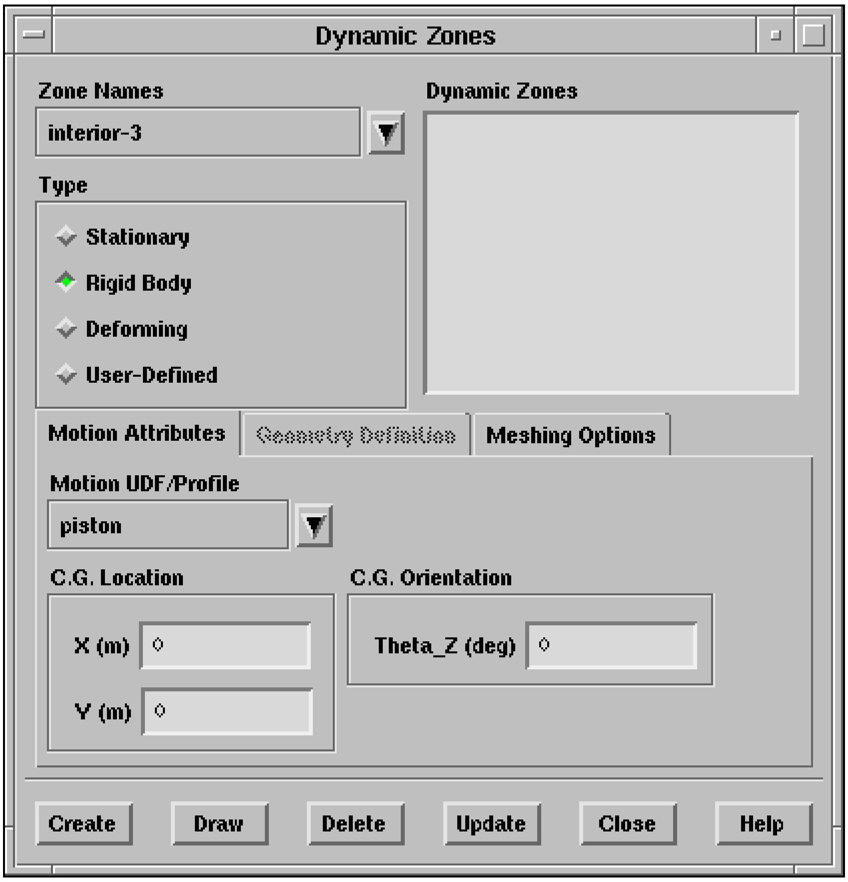


图 18.3.7.1：Dynamic Zones 面板

在 **Dynamic Zones** 面板（如图 18.7.1 所示）中的 **Type** 的下面选择 **Rigid Body**，点击 **Motion Attributes** 标签。最后，从 **Motion UDF/Profile** 的下拉菜单中选择函数名（如，piston），然后点击 **Create**，再点击 **Close**。

2. DEFINE_GEOM

一旦通译或编译了 DEFINE_GEOM UDF，函数名将会在 FLUENT 的 **Dynamic Zones** 面板（如图 18.3.7.3 所示）中变地可视与可选。

连接 DEFINE_GEOM UDF 到 FLUENT 的步骤与 DEFINE_GEOM UDF 的相同，所不同的是，在 **Dynamic Zones** 面板（如图 18.3.7.2 所示）中的 **Type** 的下面选择 **Deforming**，点击 **Geometry Definition** 标签。从 **Definition** 下面的下拉菜单中选择 **user-defined**，**Geometry UDF** 的下拉菜单中选择函数名（如，parabola）。



图 18.3.7.5.2：Dynamic Zones 面板

3. DEFINE_GRID_MOTION

一旦通译或编译了 DEFINE_GRID_MOTION UDF，函数名将会在 FLUENT 的 **Dynamic Zones** 面板（如图 18.3.7.3 所示）中变地可视与可选。

连接 DEFINE_GRID_MOTION UDF 到 FLUENT 的步骤与 DEFINE_GEOM UDF 的相同，所不同的是，在 **Dynamic Zones** 面板（如图 18.3.7.3 所示）中的 **Type** 的下面选择 **User-Defined**，并点击 **Motion Attributes** 标签。从 **Mesh Motion UDF** 下拉菜单中选择函数名（如，beam）。

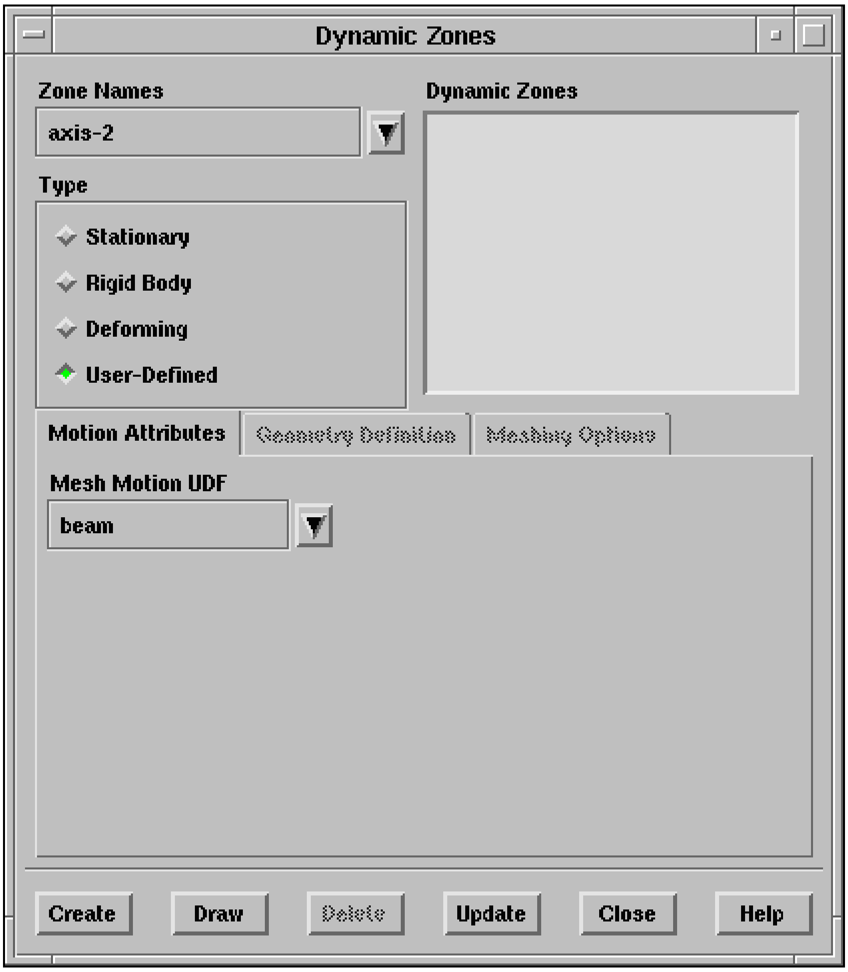


图 18.3.7.5.3：Dynamic Zones 面板

8.4.8　连接 DPM UDF 到 FLUENT

1.　DEFINE_DPM_BC

一旦通译或编译了 DEFINE_DPM_BC UDF，函数名将会在 FLUENT 的适当的边界条件面板（如图 18.3.8.1 所示）中变地可视与可选。

Define → Boundary Conditions

假设 UDF 在一个壁面上定义了一个粒子速度的边界条件。要连接 UDF 到 FLUENT，首先打开 **Wall** 边界条件的面板，选择 DPM 标签（如图 18.3.8.1）

Define → Dynamic Mesh → Zones…

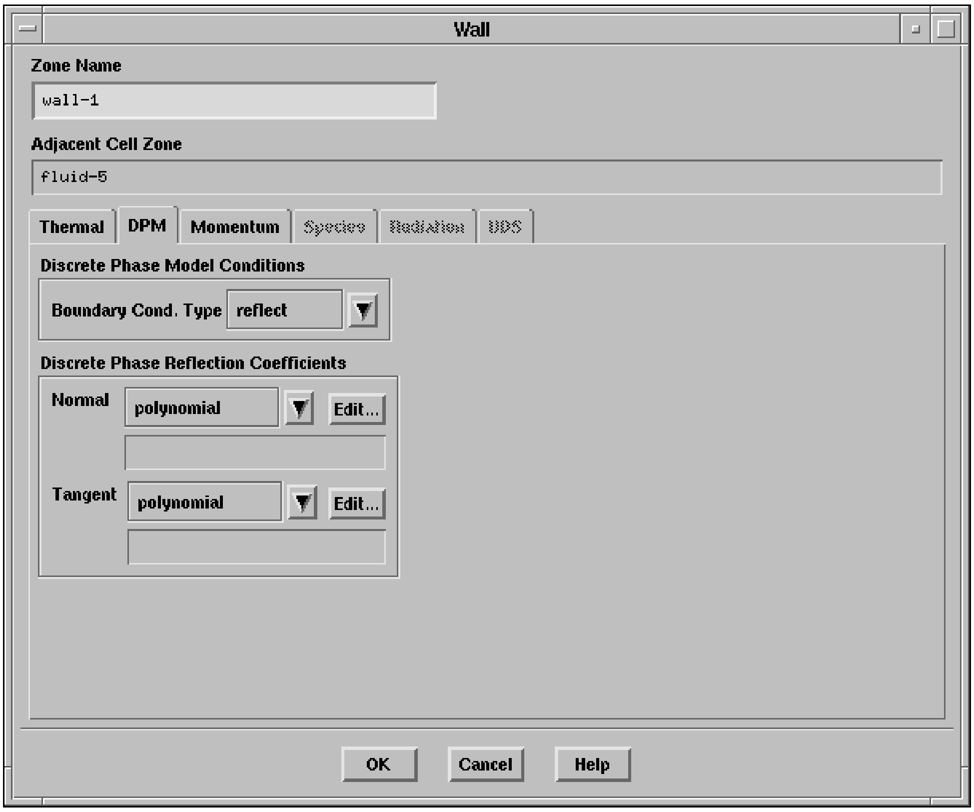


图 18.3.8.1：Wall 面板

然后，在 **Wall** 面板中（在 DPM 标签中），在 **Discrete Phase Model Conditions** 下面的 **Boundary Cond. Type** 的下拉菜单中选择的 **user_defined**，这样将打开一个面板，在这个面板中的 **Boundary Cond. Function** 的下拉菜单中选择函数名（如，reflect）。点击 **OK**。

2. DEFINE_DPM_BODY_FORCE

一旦通译或编译了 DEFINE_DPM_BODY_FORCE UDF，函数名将会在 FLUENT 的 **Discrete Phase Model** 面板（如图 18.3.8.2 所示）中变地可视与可选。

Define → Models → Discrete Phase…

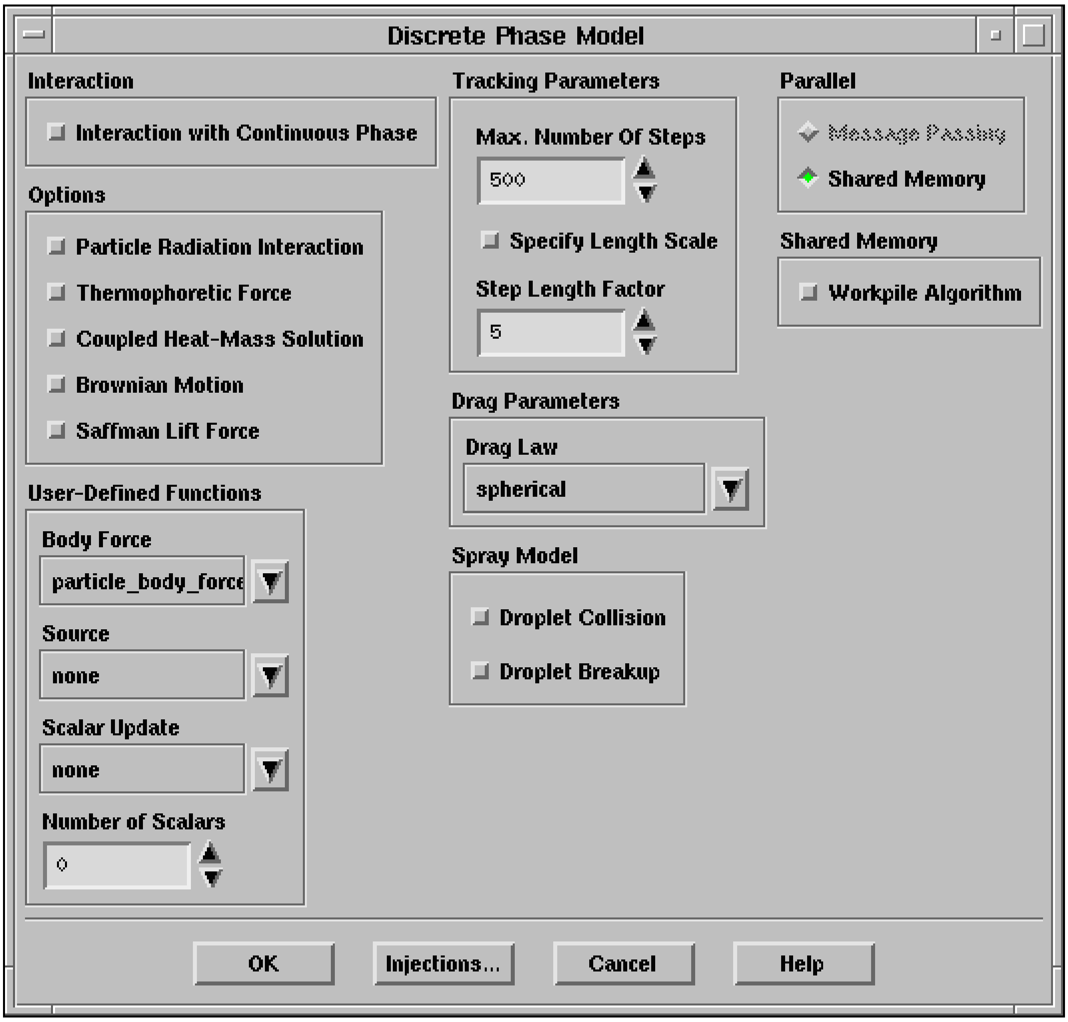


图 18.3.8.2：Discrete Phase Model 面板

要连接 UDF 到 FLUENT，在 **User-Defined Functions** 下面的 **Body Force** 下拉菜单中选择函数名（如，particle_body_force）。然后点击 **OK**。

3. DEFINE_DPM_DRAG

一旦通译或编译了 DEFINE_DPM_DRAG UDF，函数名将会在 FLUENT 的 **Discrete Phase Model** 面板（如图 18.3.8.3 所示）中变地可视与可选。

Define → Models → Discrete Phase…

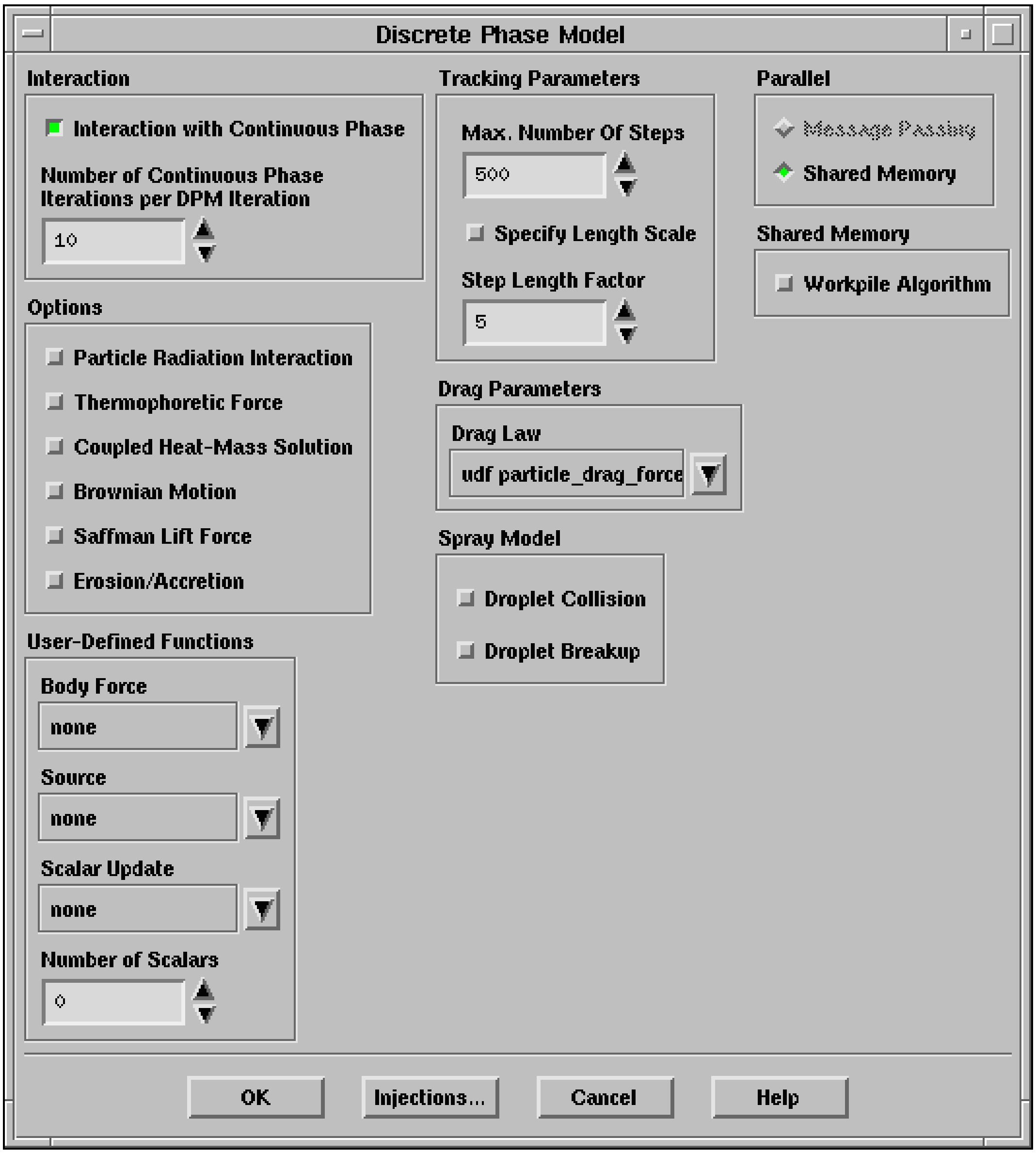


图 18.3.8.3：Discrete Phase Model 面板

要连接 UDF 到 FLUENT，在 **Drag Parameters** 下面的 **Drag Law** 下拉菜单中选择函数名（如，particle_drag_force）。然后点击 **OK**。请注意，在下拉菜单中的函数名前面有单词 **udf**，如 udf particle_drag_force。

4. DEFINE_DPM_EROSION

一旦通译或编译了 DEFINE_DPM_EROSION UDF，函数名将会在 FLUENT 的 **Discrete Phase Model** 面板（如图 18.3.8.3 所示）中变地可视与可选。

Define → Models → Discrete Phase…

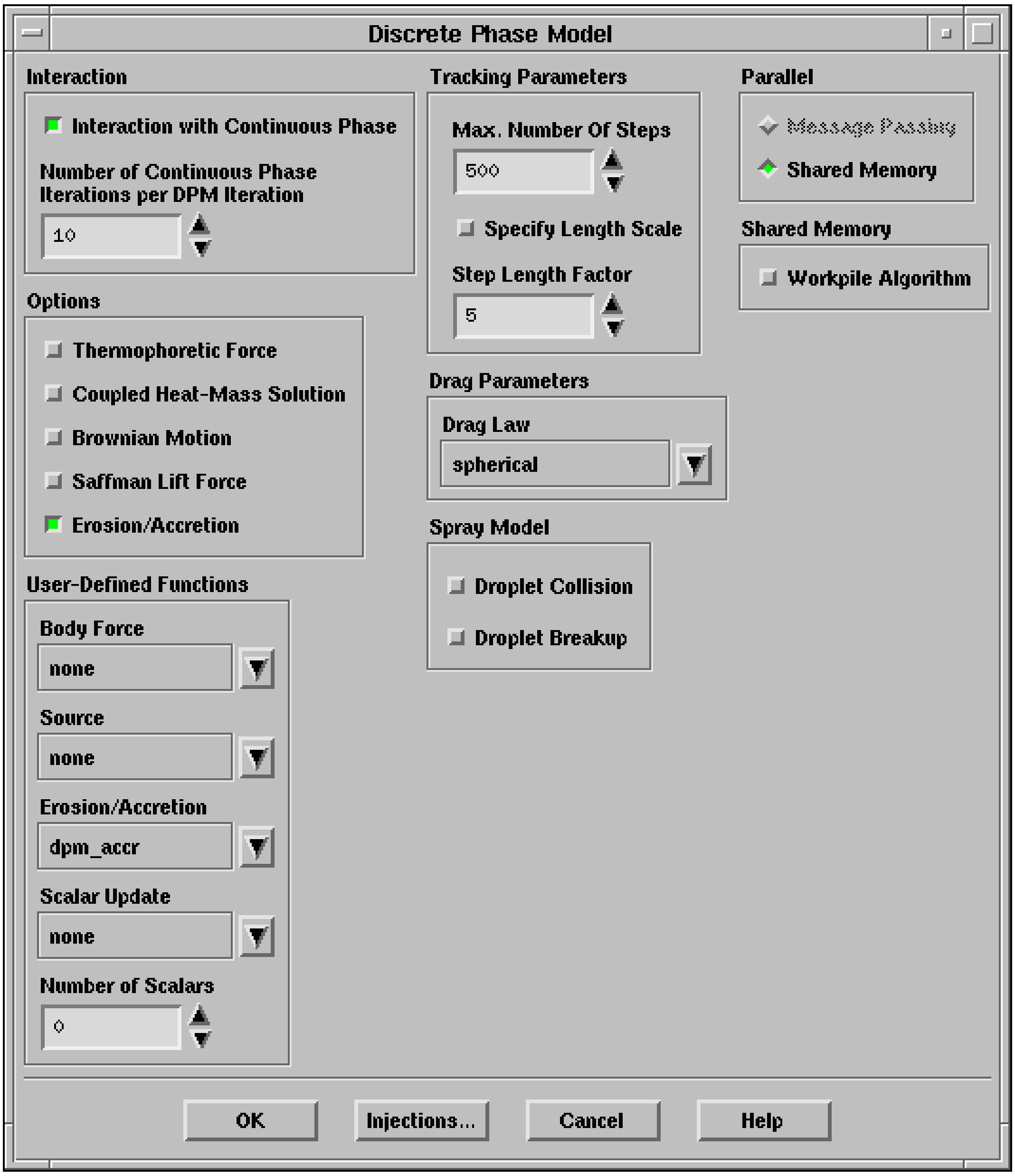


图 18.3.8.4：Discrete Phase Model 面板

要连接 UDF 到 FLUENT，激活 **Interaction** 下面的 **Interaction with Continuous Phase** 选项，然后打开 **Option** 下面的 **Erosion/Accretion**。最后，在 User-Defined Function 下面的 Erosion/Accretion 的下拉菜单中选择函数名（如，dpm_accr），点击 **OK**。

5. DEFINE_DPM_INJECTION_INIT

一旦通译或编译了 DEFINE_DPM_INJECTION_INIT UDF，函数名将会在 FLUENT 的 **Set Injection Properties** 面板（如图 18.3.8.5 所示）中变地可视与可选。

Define → Injections…



图 18.3.8.5：Set Injection Porperties 面板

在连接 UDF 之前，需要在 **Injections** 面板中建立粒子注射。在 **Injections** 面板中点击 **Create** 打开 **Set Injection Properties** 面板，建立粒子注射。

下面连接 UDF 到 FLUENT。在 **Set Injection Properties** 面板中选择 **UDF** 标签，然后在 **User-Defined Functions** 下面的 **Initialization** 下拉菜单中选择函数名（如，init_bubbles）。点击 **OK**。

6. DEFINE_DPM_LAW

一旦通译或编译了 DEFINE_DPM_LAW UDF，函数名将会在 FLUENT 的 **Custom Laws** 面板（如图 18.3.8.6 所示）中变地可视与可选。要连接 UDF 到 FLUENT，首先点击 **Injections** 面板中的 **Create** 打开 **Set Injection Properties** 面板。

Define → Injections…

其次，在 **Set Injection Properties** 面板中，点击 **Custom** 选项，这会打开 **Custom Laws** 面板。



图 18.3.8.6：Custom Laws 面板

最后，在 **Custom Laws** 面板中，从六个粒子法则中的某个适当的（例如 **First Law**）下拉菜单中选择函数名（例如，custom_law），然后点击 **OK**。

7. DEFINE_DPM_OUTPUT

一旦通译或编译了 DEFINE_DPM_OUTPUT UDF，函数名将会在 FLUENT 的 **Sample Trajectories** 面板（如图 18.3.8.7 所示）中变地可视与可选。

Report → Discrete Phase → Sample…

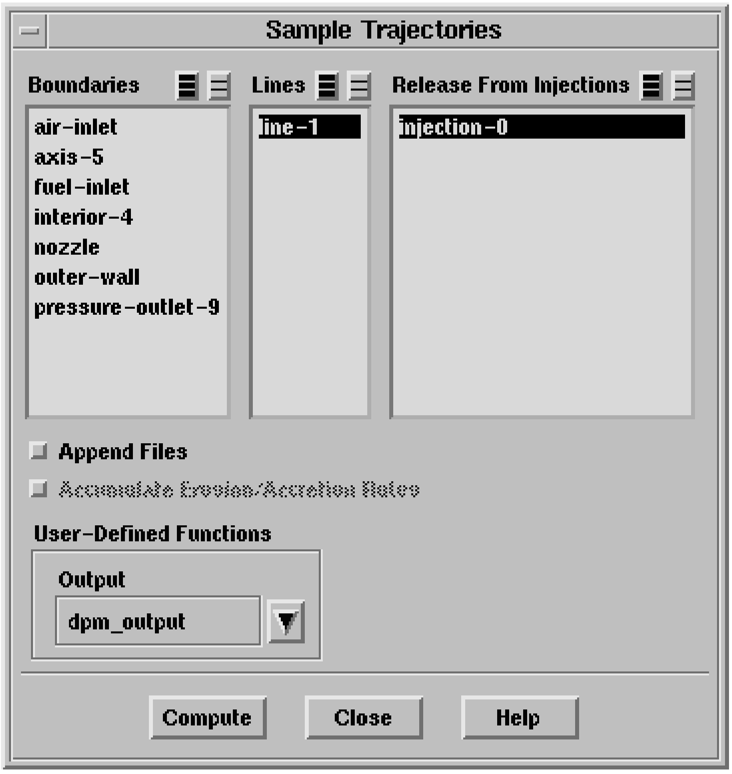


图 18.3.8.7：Sample Trajectories 面板

要连接 UDF 到 FLUENT，在 **User-Defined Functions** 下面的 **Output** 下拉菜单中选择函数名（如，dpm_output）。然后点击 **Compute** 和 **Close**。

8. DEFINE_DPM_PROPERTY

一旦通译或编译了 DEFINE_DPM_PROPERTY UDF，函数名将会在 FLUENT 的 **User-Defined Functions** 面板（如图 18.3.8.9 所示）中变地可视与可选。要连接 UDF 到 FLUENT，首先需打开 **User-Defined Functions** 面板，途径是在 **Materials** 面板中的适当特性（例如，Particle Emissivity）的下拉菜单中选择 **user-defined**。

Define → Materials…

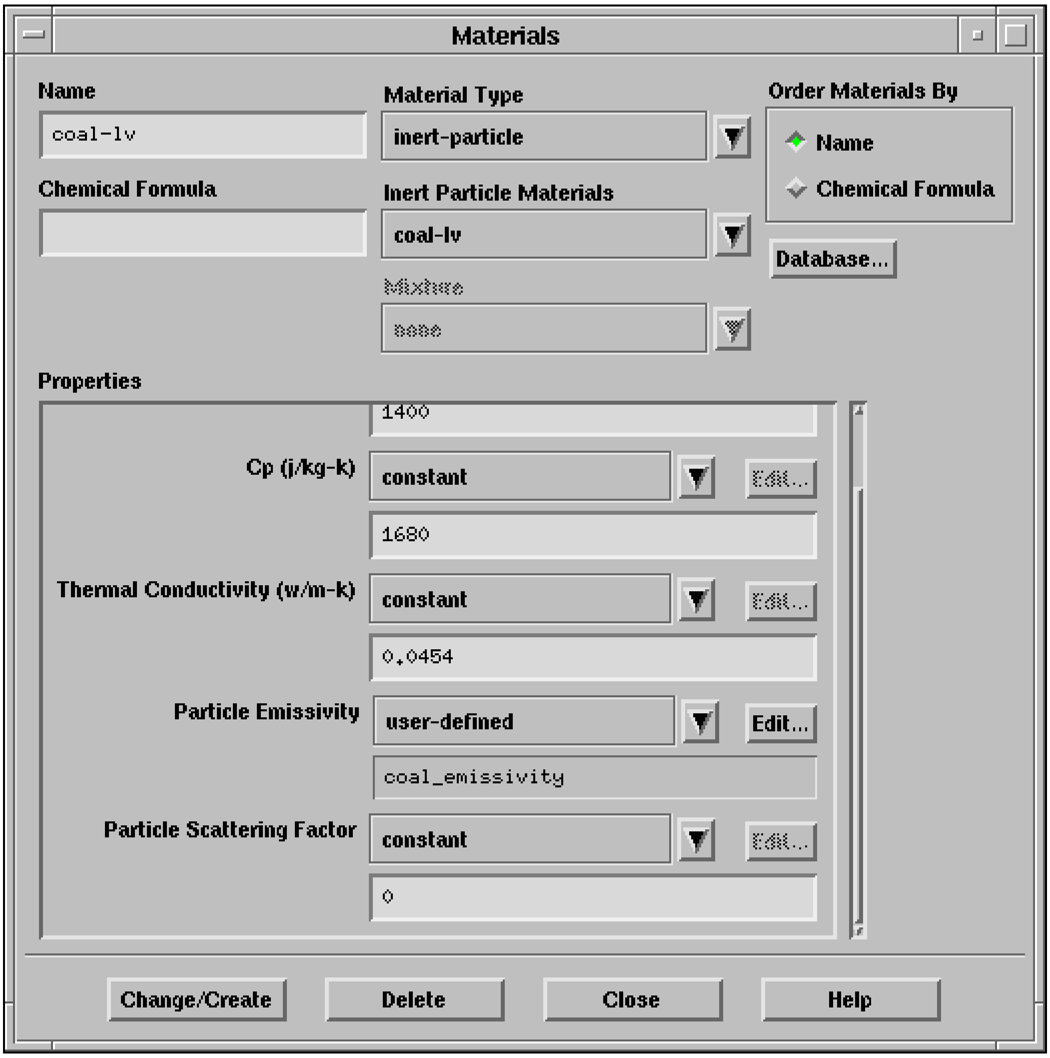


图 18.3.8.8：Materials 面板

！为了使关于 **Paticle Emissivity** 的特性显示在上述的面板中，必须激活辐射模型，在 **Discrete Phase Model** 面板中打开 **Particle Radiation Interaction** 选项，以及在 **Injections** 面板中引入一个粒子注射。

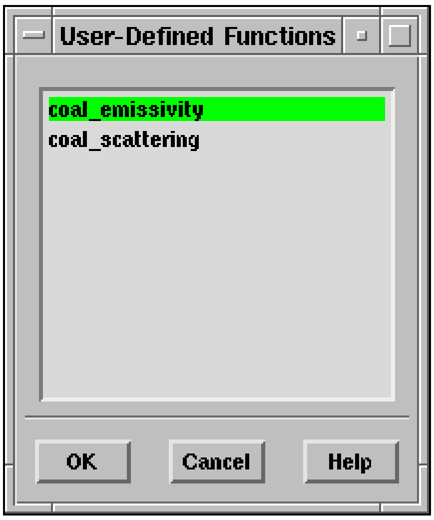


图 18.3.8.9：User-Defined Functions 面板

其次，在 **User-Defined Functions** 面板中，从显示在 UDFs 的列中选择函数名（如，coal_emissivity），点击 **OK**。随后，函数名显示在 Materials 面板中所选特性（如，Particle Emissivity）的下面。

9. DEFINE_DPM_SCALAR_UPDATE

一旦通译或编译了 DEFINE_DPM_SCALAR_UPDATE UDF，函数名将会在 FLUENT 的 **Discrete Phase Model** 面板（如图 18.3.8.10 所示）中变地可视与可选。

Define → Models → Discrete Phase…

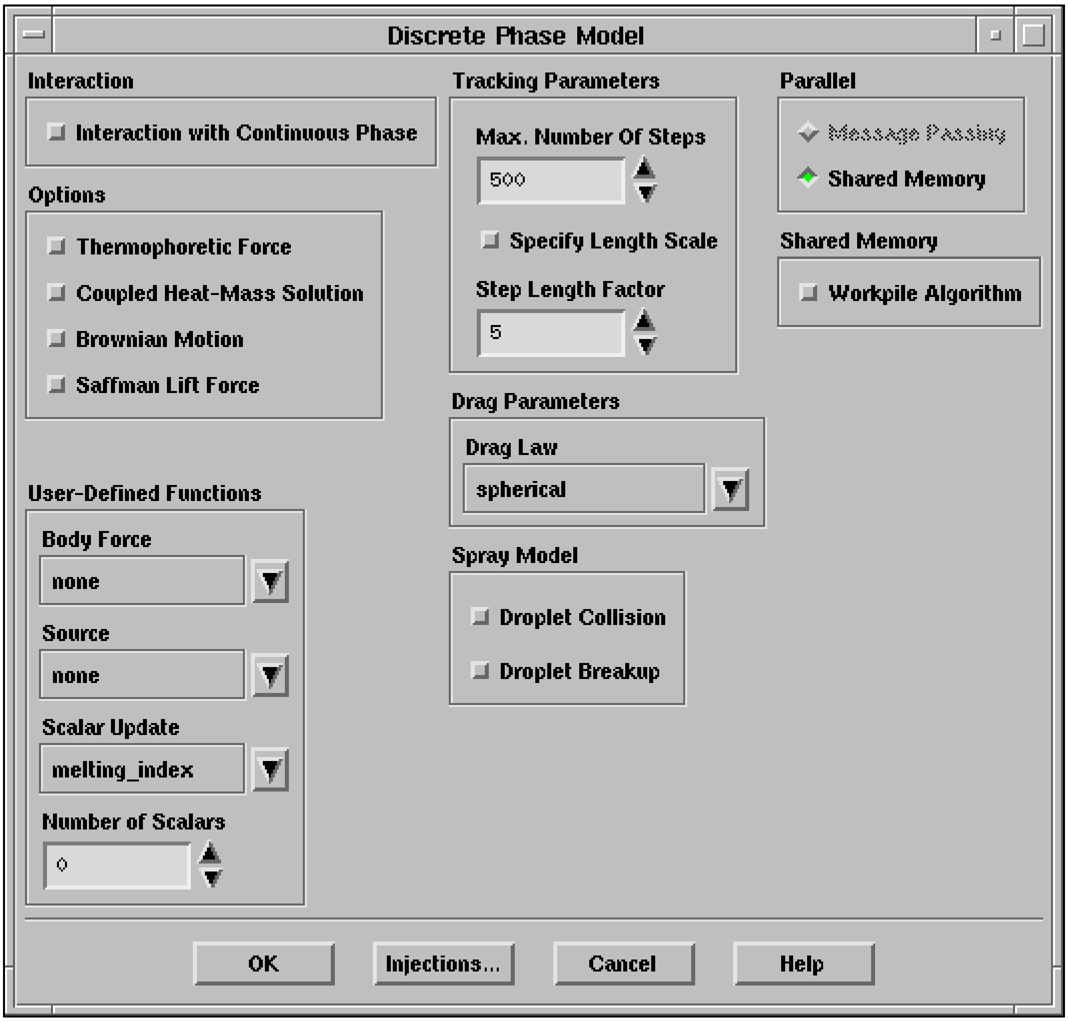


图 18.3.8.10：Discrete Phase Model 面板

要连接 UDF 到 FLUENT，在 **User-Defined Function** 下面的 **Scalar Update** 的下拉菜单中选择函数名（如，melting_index）（如图 18.3.8.10），然后点击 **OK**。

10. DEFINE_DPM_SOURCE

一旦通译或编译了 DEFINE_DPM_SOURCE UDF，函数名将会在 FLUENT 的 **Discrete Phase Model** 面板（如图 18.3.8.11 所示）中变地可视与可选。

Define → Models → Discrete Phase…

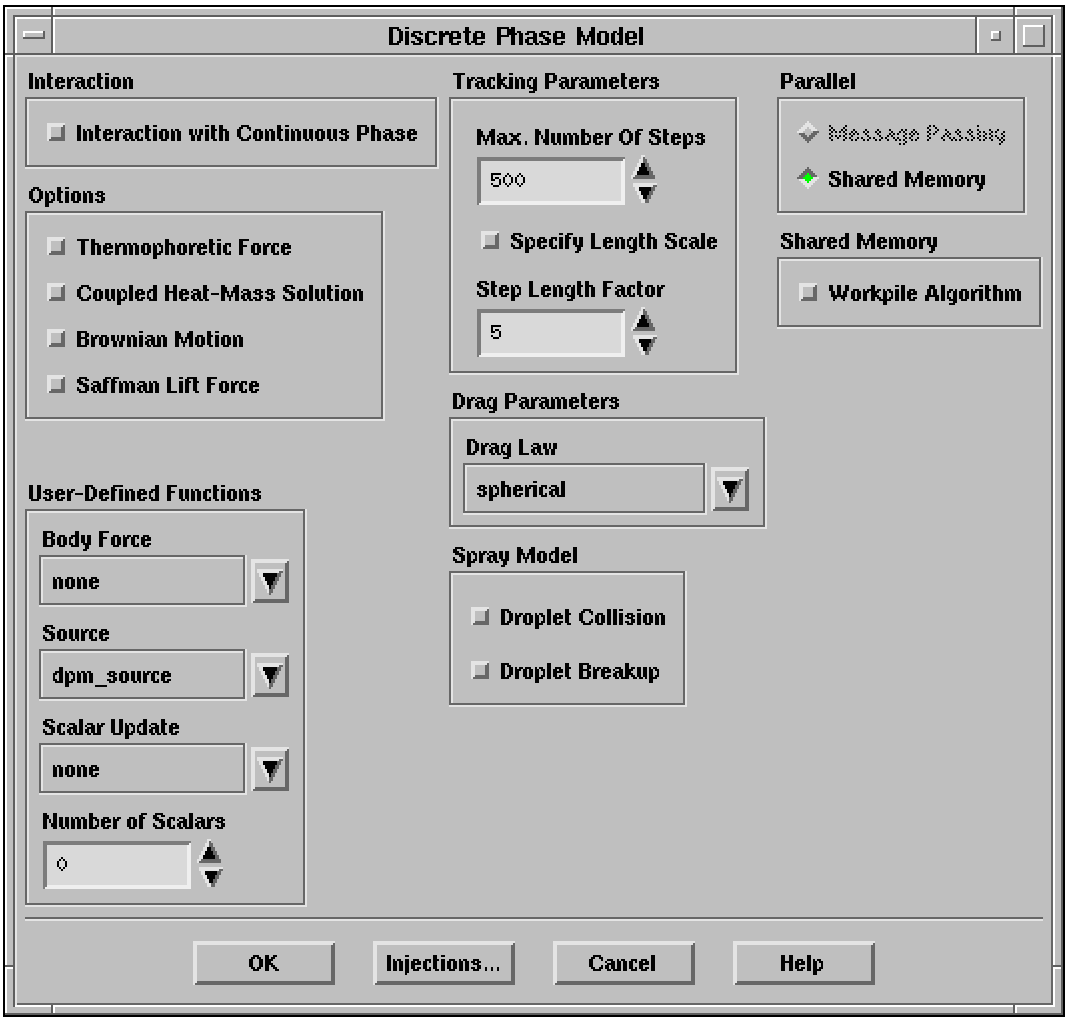


图 18.3.8.11：Discrete Phase Model 面板

要连接 UDF 到 FLUENT，在 **User-Defined Function** 下面的 **Source** 的下拉菜单中选择函数名（如，dpm_source），然后点击 **OK**。

11. DEFINE_DPM_SPRAY_COLLIDE

一旦通译或编译了 DEFINE_DPM_SPRAY_COLLIDE UDF，函数名将会在 FLUENT 的 **User-Defined Function Hooks** 面板中（如图 18.3.8.12 所示）变地可视与可选。

Define → User-Defined → Function Hooks…

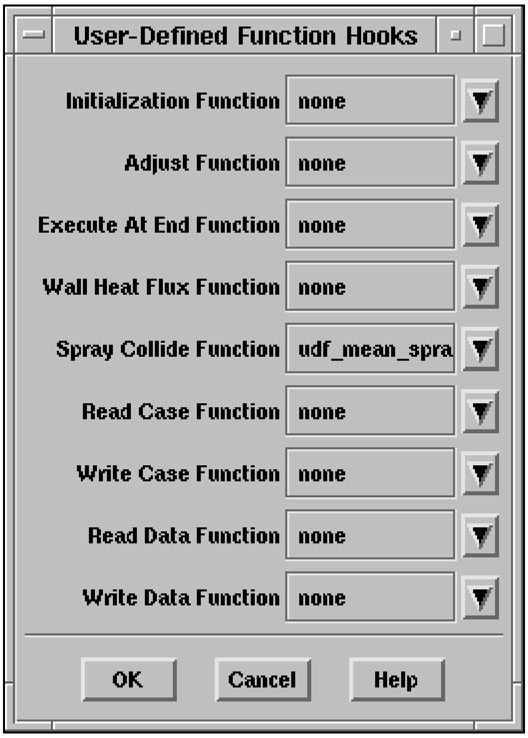


图 18.3.5.12：User-Defined Function Hooks 面板

！在连接 UDF 之前，需要激活 **Discrete Phase Model** 面板中的一个离散相模型。

要连接 UDF 到 FLUENT，在 **User-Defined Function Hooks** 面板中的 **Spray Collide Function** 的下拉菜单中选择函数名（如，udf_mean_spray），然后点击 **OK**。

12.　　DEFINE_DPM_SWITCH

一旦通译或编译了 DEFINE_DPM_SWITCH UDF，函数名将会在 FLUENT 的 **Custom Laws** 面板中变地可视与可选（如图 18.3.8.13 所示）。要连接 UDF 到 FLUENT，首先点击 **Injections** 面板中的 **Create** 打开 **Set Injection Properties** 面板。

Define → Injections…

其次，在 **Set Injection Properties** 面板中，点击 **Custom** 选项，这将会打开 **Custom Laws** 面板。



图 18.3.8.13：Custom Laws 面板

最后，在 **Custom Laws** 面板中，从 Switching 的下拉菜单中选择函数名（例如，dpm_switch）(如图 18.3.8.13)，然后点击 **OK**。

18.4 用户定义函数（UDF）举例

本章将通过一个例子按部就班的介绍一个 UDF 的编辑和它在 FLUENT 模型中的有效应用。编辑这个 UDF 总共有 7 步：

- 定义问题
- 建立一个 C 语言源代码文件
- 启动 FLUENT，读入（或建立）case 文件
- 通译源文件
- 连接 UDF 到 FLUENT
- 运行计算
- 分析数值结果及比较数值结果与期望结果

概述

作为程序的第一步，有必要应用一个 UDF 给出所希望求解的问题的定义。例如，假设想使用一个 UDF 为自己的问题定义一个自定义边界剖面，那么首先需要定义出描述这个剖面的数学方程，即建立问题的数学模型。

第二步，将需要把数学方程翻译成一个用 C 语言书写的函数。可以使用任何文本编辑器进行此工作。在工作目录中用扩展名为 a.c（例如，velocity_profile.c）的形式保存文件。

一旦写了 C 函数，准备启动 FLUENT 及读入（或建立）case 文件，这是程序的第三步。然后，通译或编译并装载 UDF 的源文件，调试这个源文件（第四步），然后连接 UDF 到 FLUENT 中（第五步）。最后，运行这个计算（第六步），分析计算模拟结果，并把模拟结果与期望的结果比较（第七步）。下面将通过一个涡轮翼型的例子一步步的介绍这个过程。

第一步：问题的定义

创建一个 UDF 的第一步是建立问题的数学模型。

图 18.4.1 是涡轮翼型的图解。翼型周围的流场模型应用非结构网格。域从一个底部的周期边界延伸到同一个边界的顶部，左边是速度进口，右边是压力出口。

图 18.4.1:涡轮翼形的网格

进口处 x 速度大小为常数 $20m/s$，图 18.4.2，图 18.4.3 分别是计算所得的速度等值线图和速度矢量图。可以看到，初始的定常速度场随着气流绕涡轮翼的流动而被曲解。为解决此问题，对入流速度进行修正，现在假设涡轮翼型的进口处的 x 速度是非一致的，可表示如下：

$$v_x = 20 - 20\left(\frac{y}{0.0745}\right)^2$$

其中，变量 y 在进口的中心取 0 值，从进口上部到下部呈线性取值，变化范围是 ± 0.0745。所以 x 速度在进口中心的速度为 $20m/s$，底边的速度为 0。

对进口速度修正后的涡轮翼进行计算所得的结果将与进口处为常数速度的计算结果进行比较。

图 18.4.2: 关于进口 x 速度为常数的速度等值线图

图 18.4.3:关于进口 x 速度为常数的速度矢量图

第二步：创建一个 C 源文件

既然已决定用这个方程定义 UDF，那么使用任一种文本编辑器去建立一个含 C 代码的文本文件。在工作目录中保存扩展名为.c 的源代码文件（例如，udfexample.c）。下面的源代码清单仅包含一个单独的 UDF。事实上，源文件中可以含有多个串联的函数。

下面是关于在一个 UDF 中怎样执行上面方程的一个例子。UDF 的功能是由主要的

DEFINE 宏指定的。这里， DEFINE_PROFILE 宏用来说明一个求解器，这个求解器中有下面代码将提供的边界上的翼型信息。其它的 DEFINE 宏将在后面讨论。

```
    /*****************************************************************
******udfexample.c
    UDF  规定一个定常速度的翼型的边界条件
    *****************************************************************
*****/

    #include”udf.h”  /*必须在每一个你写的 UDF 的开始*/
    DEDINE_PROFILE(inlet_x_velocity, thread, index)
    {
      real x[ND_ND];  /*这个将取位置矢量*/
      real y;
      face_t f;

      begin_f_loop(f, thread)  /*跳过线程中的所有的面，被传送到 DEFINE 宏自变量
中*/
       {
         F_CENTROID(x, f, thread);
         Y=x[1];
         F_PROFILE(f, thread, index) = 20.- y*y/(.0745*.0745)*20.;
       }
      end_f_loop(f, thread)
    }
```

DEFINE_PROFILE 宏的第一个自变量，inlet_x_velocity，是用来鉴别 **Velocity Inlet** 面板中的函数。自变量的名字由自己任意规定。函数中的方程将应用在一个给定的边界区域（由 thread 识别）上的所有单元面（在面环中用 f 识别）。在 FLUENT 图形用户界面里，当为一个特殊的边界选择 UDF 时，线程是被自动定义的。指针是通过 begin_f_loop 设备被自动定义的。在这个 UDF 中，begin_f_loop 宏用来跳过边界区域中的所有单元面。对于每一个面，面的中心坐标是通过 F_CENTROID 宏存取的。

y 坐标的 **y** 用于比喻的剖面方程中，返回的速度通过 F_PROFILE 宏分配到这个面上。Begin_f_loop 和 F_PROFILE 是 Fluent 提供的宏。

第三步：启动 FLUENT 和读（或建立）Case 文件

一旦创建了 UDF，准备在 FLUENT 中开始建立问题。

1. 从工作的目录中启动 FLUENT。
2. 读（或建立）case 文件。

第四步：通译源文件

现在准备通译在第二步中创建的 UDF，并且 UDF 包含在名为 udfexample.c 的源文件中。注意，这一步不适合用 Windows 并行网络。

1. 打开 Interpreted UDFs 面板。

Define → User-Defined → Functions → Interpreted…

图 18.4.4：Interpreted UDFs 面板

2. 规定源文件，其中有两种途径：直接在 **Select File** 面板中的 **Source File Name** 下面的文本框中键入名字，另外一种方法是点击面板中的 **Browse…**，从合适的目录中选择 UDF 的源文件名。

3. 在 CPP Command Name 下的框里规定用 C 预处理器。可以保留默认的 cpp，也可以选择 Use Contributed CPP 使用 Fluent.Inc 提供的预处理器。

4. 保留 **Stack Size** 里的默认值 10000，除非函数中的局部变量数会引起堆叠从而导致溢出。在这种情况下，在 **Stack Size** 里设置的数要大于局部变量使用的数。

当函数在翻译时，如果希望一列编译语言代码出现在自己的控制台窗口中，那么点击 **Display Assembly Listing**。

点击 **Interpret** 通译 UDF。

```
Inlet_x_velocity:
                    .local.pointer thread (r0)
                    .local.int nv (r1)
  0                 .local.end
  0                 save
                    local.int f (r3)
                         0    push.int 0
                         .local.pointer x (r4)
                         3    begin.data 8 bytes, 0 bytes initialized:
                         7    save
         .               .
         .               .
         .               .
         156             pre.inc.int f (r3)
         158             pop.int
         159             b .L3 (22)
         .L2
                         161  restore
                         162  restore
                         163  ret.v
```

! 注意,如果编辑失败,那么 FLUENT 将报告一个错误，那么需要对程序进行调试。

当通译完成后，点击 **Close** 按扭。

写 case 文件。名为 inlet_x_velocity 的通译的 UDF，将被保存 case 文件中，以便无论何时读这个 case 文件时，函数都将被自动地翻译。如果选了 **Display Assembly Listing** 项，那么编译代码将在控制台窗口中出现。

! 除了翻译关于剖面 UDF 的这一特殊的例子，也可以使用 FLUENT 中 **Compiled UDFs** 面板编译 UDF。编译 UDFs 的详细内容在第 18.3 节中介绍。

第五步：连接 UDF 到 FLUNET

第四步工作即 UDF 通译完成后，开始下一步工作：使用一个图形用户界面面板连接这个函数到 FLUENT。对这个特殊剖面 UDF 的例子，这个函数的名字将出现在 **Velocity Inlet**

的边界条件面板中的 **Zone Name** 下面的文本框里。

Define → Boundary Conditions…

在 **X-Velocity** 下面的列中，选择 **udf inlet_x_velocity**，它是我们样本例题中函数的名字。一旦被选择，这个 UDF 就会在计算中使用，而不是出现在 X-Velocity 框里面的那个 0 值（在这个例题中）。若接受新的边界条件，点击 **OK**，然后关掉这个面板。

第六步：运行计算

和通常一样，运行这个计算。

Solve → Iterate…

第七步：分析数值结果及比较数值结果与期望结果

一旦计算结果收敛，可以获得修正后的速度场。图 18.4.5 给出了比喻的进口 x 速度的速度等值线图，可以与 $20m/s$ 的常数速度的结果（图 18.4.2）比较一下。对于常值速度情况，流场随着气流围绕涡轮翼的流动而受到曲解。然而，对于这个施以比喻的剖面的速度场在入口的中心呈现一个最大值，到达入口的边上则降到了 0。

图 18.4.5：关于比喻的进口 x 速度的速度等值线图